Multiresolution Image Segmentation

Abhir H. Bhalerao B.Sc.

A thesis submitted to
The University of Warwick
for the degree of
Doctor of Philosophy

November 1991

Multiresolution Image Segmentation

Abhir H. Bhalerao B.Sc.

A thesis submitted to
The University of Warwick
for the degree of
Doctor of Philosophy

November 1991

Summary

Image segmentation is an important area in the general field of image processing and computer vision. It is a fundamental part of the 'low level' aspects of computer vision and has many practical applications such as in medical imaging, industrial automation and satellite imagery. Traditional methods for image segmentation have approached the problem either from localisation in class space using region information, or from localisation in position, using edge or boundary information. More recently, however, attempts have been made to combine both region and boundary information in order to overcome the inherent limitations of using either approach alone.

In this thesis, a new approach to image segmentation is presented that integrates region and boundary information within a multiresolution framework. The role of uncertainty is described, which imposes a limit on the simultaneous localisation in both class and position space. It is shown how a multiresolution approach allows the trade-off between position and class resolution and ensures both robustness in noise and efficiency of computation.

The segmentation is based on an image model derived from a general class of multiresolution signal models, which incorporates both region and boundary features. A four stage algorithm is described consisting of: generation of a low-pass pyramid, separate region and boundary estimation processes and an integration strategy. Both the region and boundary processes consist of scale-selection, creation of adjacency graphs, and iterative estimation within a general framework of maximum *a posteriori* (MAP) estimation and decision theory. Parameter estimation is performed in situ, and the decision processes are both flexible and spatially local, thus avoiding assumptions about global homogeneity or size and number of regions which characterise some of the earlier algorithms. A method for robust estimation of edge orientation and position is described which addresses the problem in the form of a multiresolution minimum mean square error (MMSE) estimation. The method effectively uses the spatial consistency of output of small kernel gradient operators from different scales to produce more reliable edge position and orientation and is effective at extracting boundary orientations from data with low signal-to-noise ratios.

Segmentation results are presented for a number of synthetic and natural images which show the cooperative method to give accurate segmentations at low signal-to-noise ratios (0 dB) and to be more effective than previous methods at capturing complex region shapes.

Key Words:
Segmentation, Multiresolution, MAP Estimation, Orientation

Contents

List of Figures

List of Tables

Notation used in this thesis

$x_{ij}(l)$	Grey level data at pixel (i, j) on pyramid level l
$0 \leq l \leq M$	Pyramid level l, root level 0, image level M
A_{mn}	Filter kernel (matrix)
$\bigcup$	Union
$\bigcap$	Intersection
$\vec{\xi}_k = (i, j, l)^T$	Scale-space position vector (region)
$\vec{\chi}_k = (i, j, l)^T$	Scale-space position vector (boundary)
Λ_i	Index set
ξ_0, ξ_1, ξ_2	Vector components
$\mathcal{N}_{\vec{\xi}}$, $\mathcal{L}_{\vec{\xi}}$	Neighbourhood sets
card$\{X\}$	Cardinality of set X
$P(x)$	Probability of x
$p(x)$	Probability density function of x
H_0, H_1	Hypotheses
$E[x]$	Expected value of x
var$x$	Variance of x
$\mathbf{v} = (v_0, v_1)^T$	Vector data in bold face
arg$(\mathbf{v})$	Argument of 2d vector $(\tan^{-1}(\frac{v_1}{v_0}))$
$\lvert\mathbf{v}\rvert$	Length of vector $(\sqrt{v_0^2 + v_1^2})$
$\exp(x)$	Exponential function e^x
$\lceil x \rceil$	Ceiling function

Acknowledgements

This work was funded by UK SERC and by Shell Research Ltd., and conducted within the Image and Signal Processing Research Group in the Department of Computer Science at Warwick University.

I would like to thank all the staff at both the Computer Science Department and Shell Research. In particular, thanks go to all my friends and colleagues, past and present, of the Image and Signal Processing Group at Warwick: Andrew Calway, Simon Clippingdale, Roddy McColl, Andrew Davies, Wooi Boon Goh, Matthew Jackson, Edward Pearson, Martin Todd, June Wong and Horn-Zhang Yang. They have made numerous contributions and have provided a stimulating environment in which to work. Thanks also to Jeff Smith for providing essential software support, and the staff of the University Photographic Department for their help with the photographs.

I would also like to thank my supervisor at Shell Research, Dr. Steve Graham for his enthusiasm and support.

I am particularly indebted to my supervisor Dr. Roland Wilson, without whose ideas, enthusiasm and profound expertise in the subject this work would not have been possible.

This work is dedicated to my mother and, in loving memory, to my father.

Chapter 1

Introduction

1.1 Seeing and Computer Vision

Of all the perceptual mechanisms that we possess, vision is undisputably the most important. The subconscious way that we often look, interpret and ultimately act upon what we see, however, belies the complexity of visual perception. The comparatively young science of vision research is aimed at the understanding of the general problem of seeing. The automation of the task, by the use of image capture equipment in place of our eyes, and computers and algorithms, in place of the little understood visual 'wetware', constitutes what is termed computer vision [5] . The Human Visual System (HVS) is an important model for any work in vision because it is clearly both efficient and general purpose, which are also the goals of any computer vision system.

Computer vision has obvious applications and benefits. Examples range from medical image processing, where quantification, identification and diagnosis are often the aims, to satellite imagery for, say, crop analysis, or restoration of astronomical images [82] [41] [84]. The applications are not limited to single image frames, since with the advances in computer technologies, it is becoming possible to tackle stereo images and perform analysis of image volumes and sequences. In the area of robotics, vision becomes an important part of the sensory inputs used to control the motor responses [35]. In fact, anywhere a human observer

can be replaced by a machine may be regarded as an application suitable for computer vision. With the growth of image sources and as the boundaries of vision are expanded, at both the microscopic scale, such as in electron microscopy, and the macroscopic scale, seeing ever more remotely into the universe, not to mention the explosion in the availability and use of electronic visual media, the solution of the vision problem becomes ever more pertinent.

1.2 The Segmentation Problem

Image segmentation is one part of the general task of computer vision. A 'common sense' definition would describe segmentation as the translation of an image, from an array of grey levels, to a symbolic description, for example as a number of well defined regions. Given the artificial image shown in figure 1.1(a), the segmentation is the seemingly simple task of separating the foreground from the background while ignoring the presence of noise. In this image the inter-regional signal to noise ratio is 1, and yet it does not present a significant problem to us. There are two related problems in the process of segmenting this image: first, detecting the presence of 5 objects; secondly, the harder problem of locating their boundaries [109].

In the classical pattern recognition paradigm, image segmentation could be described as combined feature selection and classification, which is indeed the basis of much of the work in this area [36] [92] [33]. An alternative way to regard segmentation is to consider it in terms of the computer vision model, where the regions of interest in the image are the projections of the 'objects' in the scene and then the task of segmentation is to identify and locate these objects, regardless of the indeterminancy due to image acquisition, lighting and so on. In figure 1.1(b), for example, we are easily able to segment out the hat, or the edge of the mirror, which although they are 3-dimensional entities in the real world, project onto the image plane as 2-d regions.

(a) (b)

Figure 1.1: 'shapes' image (SNR = 0dB), 'girl' image, (256 × 256)

The processing of information in the HVS is known to be hierarchical [98]. Low level vision is concerned with processing the signal level of the 'retinal image' [51] to a simple representational level. High level vision, which is less understood, is concerned with interpretation of the low level information. As a processing step of the HVS, segmentation is certainly a 'low level' task, which extracts and groups together perceptually important features of the scene. Segmentation does contain, however, the beginnings of domain-dependent interpretation [5]; the question is in what form is this *prior* information represented and to what degree is it used. Returning to the synthetic image of figure 1.1(a), one interesting question is to what extent is our prodigious performance at identifying the shapes explained by our familiarity with them as circles, triangles etc. An example to test this, which will be used as a test image in this work (see Chapter 4), consists of a set of random boundary 'widgets', which we cannot have shape knowledge about, but nevertheless can still segment.

Another important aspect of the segmentation process is its invariance to various transformations of the signal, such as translation or rotation, or changes in illumination. It

is important, therefore, that such invariance is incorporated into any scheme for image segmentation.

There is now a great deal of argument and debate on the merits of a model-based approach to vision, such as that proposed and advocated by Marr [68]. Model-based vision extrapolates the low level representational structures, through a hierarchy of increasing abstraction, whereby interpretation is achieved by matching against some model of the world. These ideas are being found to be too restrictive in the philosophy of 'active vision' and the like [3]. But what of low level vision? It is by no means solved and examples such as figure 1.1(a) demonstrate how powerful the low level visual processes are. The challenge to an automated segmentation system is to achieve comparable performance at modest computational cost.

1.3 Uncertainty and Multiresolution Analysis

The task of segmentation, and image analysis in general, may be described as an inference process: given the image intensities in a neighbourhood of X, an elementary event of class C is (believed to be) located at position X [104] [105]. For example, in the case of figure 1.1(a), to determine the classes of the objects some form of local average grey level can be used. However, to be more certain of the class of a particular pixel requires averaging over a larger area, which in turn makes the location of the object boundaries less certain. Thus the conjunction of the two propositions, "an event of class C" *and* "located at position X", are inextricably tied through uncertainty. In other words, localisation in class space conflicts directly with the simultaneous localisation in position space.

In multiresolution methods an image is analysed over several resolutions or scales. This is equivalent to trading off high position resolution, present at the image plane, with obtaining greater class resolution at lower scales (figure 1.2). This is only half the story, of course, as

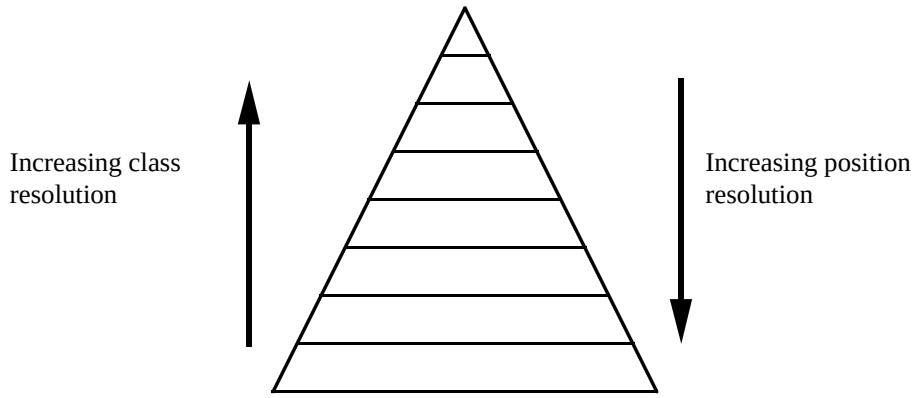


Figure 1.2: Class/position resolution trade-off

there is still the problem of how to combine the information across scales, i.e. to regain the spatial resolution which has been lost.

In the inference process of determining 'what is where', 'what' is an 'object', the presence of which is established in some appropriate property domain, and the 'where' gives its location in the spatial domain [68] [109]. Consider the image signal $x_{\vec{\zeta}}, \vec{\zeta} \in \mathcal{N}_{\vec{\xi}}$, in the neighbourhood of the point $\vec{\xi}$. The problem is to determine the presence of an object C at that point and estimate its parameters $\phi^C_{\vec{\xi}}$. This can be cast into statistical terms, based on the posterior probabilities of the parameters and the hypothesis $H^C_{\vec{\xi}}$ that there is an object C at $\vec{\xi}$ [39] [100]

$$P(\phi^C_{\vec{\xi}}, H^C_{\vec{\xi}} \mid x_{\vec{\zeta}}, \vec{\zeta} \in \mathcal{N}_{\vec{\xi}}) =$$
$$\text{Prob}\left\{C \text{ at } \vec{\xi} \text{ with parameters } \phi^C_{\vec{\xi}}, \text{ given } x_{\vec{\zeta}}, \vec{\zeta} \in \mathcal{N}_{\vec{\xi}}\right\} \tag{1.1}$$

An important idea in the work described here is that of *scale consistency* [20] [100]. This is based on the assumption that there is more than one scale where there is just one C within the neighbourhood of a given point $\vec{\xi}$. Scale consistency can be used to determine the appropriate scale of a feature by first performing the inference of equation (1.1) at each scale separately, and then checking the results for consistency between successive levels. The consistency check uses the initial estimates of the parameters at one scale to provide

an improved parameter estimate and a better assessment of the hypothesis $H^C_{\vec{\xi}}$. This step is then repeated until an acceptable level of confidence is reached. Using the formulation of (1.1) with a multiresolution representation allows the hypotheses to be fixed and the size of the otherwise unspecified neighbourhood $\mathcal{N}_{\vec{\xi}}$ to be varied.

The pyramid structure depicted in figure 1.2 is just one of a number of different multiresolution representations that have been used in image processing, and has the property that the spatial resolution is reduced by some factor (usually 2) at each scale, creating a stack of successively smaller images - hence the name pyramid [18]. The simplest type of pyramid is the quadtree [90] which, although simple to generate, has been shown to be a useful analysis structure for segmentation [88], coding [106], and modelling and estimation [26]. More sophisticated multiresolution structures, such as Wavelet representations [67] [29] and the Multiresolution Fourier Transform (MFT) [20] [30] have now been advocated as generalised multiresolution feature representations for image analysis. Moreover, the class-position trade off is only one of several useful properties that multiresolution structures have. It has been demonstrated that they can be computationally efficient and appropriate for image modelling [25] (see also Chapter 2).

1.4 A Review

A review of some notable segmentation methods used by other workers is presented in this section. The purpose is to identify the common themes in the various approaches to segmentation, to highlight their advantages and limitations, and to set a context for the work presented in this thesis. The methods are categorised into three broad classes: region based, boundary based and combined approaches. Since multiresolution structures form the basis of much of the successful work in this area, attention is concentrated on such approaches.

1.4.1 Region based approaches

Perhaps the simplest approach to segmentation is to *threshold* the grey level image. The threshold is an intensity level which is chosen to divide the image pixels into either object or background. This is clearly only possible with a very restricted set of images. For example, there are industrial 'conveyor belt' applications, where segmentation is used to determine attributes of some metal widget, such as area or orientation, and forms the 'vision' part of a simple robotic system. Under the assumptions of a controlled environment these methods, although specific, are nevertheless efficient. As methods to tackle natural images, however, where variation of illumination, noise and texture are present, they are grossly inadequate.

The thresholds are determined by some form of measurement space clustering (see [79] [41]), typically using a histogram as the class space. Finding the modes determines the partitions of the space and hence the segmentation. The histogram is 'global' and retains no positional information. The major draw back of this is that it is invariant to spatial rearrangement of the pixels, which is an important aspect of what is meant by segmentation [109]. This is not to say that a histogram cannot provide useful information, as the Spann and Wilson method [88] demonstrates (see below).

The need to incorporate some form of spatial information into the segmentation process, i.e. to achieve a degree of positional locality of the decision criteria, led to the development of methods where pixels are classified using their *context* or neighbourhood. Such techniques fall under the general heading of relaxation methods [92] [59] [44]. From an initial classification of the pixels, the class of a given pixel is iteratively updated, using a probabilistic model, with reference to the class of the neighbouring pixels. The basic premise of these context dependent classification methods is that the classes of the objects, i.e. pixels, are interrelated. Such an assumption is valid when dealing with images because the classes of neighbouring pixels are usually not independent. The context, therefore, is based on

geometric proximity.

In Relaxation Labelling (RL) [54] [76], each object $\vec{\xi}_i, 1 \leq i \leq N$ is assigned a probability $P_k(\vec{\xi}_i)$ that it belongs to class $x_k, 1 \leq k \leq N_c$. These 'probabilities' are made to lie in the range $[0, 1]$ and $\sum_{k=1}^{N_c} P_k(\vec{\xi}_i) = 1$. The contextual information is incorporated into the scheme in the form of a *compatibility function*

$$\begin{aligned} C_{kl}(i,j) = &\text{ Extent to which} \\ &\text{'}\vec{\xi}_i \text{ belongs to } x_k\text{' is compatible with '}\vec{\xi}_j \text{ belongs to } x_l\text{'} \end{aligned} \tag{1.2}$$

At iteration t the *credibility factor* due to each object $\vec{\xi}_j$ related to $\vec{\xi}_i$ is calculated by [92]

$$\lambda_k^t(i;j) = \sum_{l=1}^{N_c} C_{kl}(i,j) P_l^t(j) \tag{1.3}$$

Note that $\lambda_k^t(i;j)$ is large if the compatibility is large and the probabilities are high. The probabilities of each object are then updated by applying a *relaxation* formula, such as

$$P_k^{t+1}(i) = \frac{1}{N} \sum_{j=1, i \neq j}^{N} \left[\frac{\lambda_k^t(i;j) P_k^t(i)}{\sum_{l=1}^{N_c} \lambda_l^t(i;j) P_l^t(i)} \right] \tag{1.4}$$

where the denominator ensures that $P_k(i)$ remain in the range $[0, 1]$.

RL is justified on heuristic grounds, as the relaxation formula adjusts decision variables in a 'reasonable' way. If the compatibility is large and the probabilities high then the probability at the next iteration will increase. If the probabilities tend to either 0 or 1 then they do not change further. Convergence is not guaranteed but in practical applications only a consistent (under some criteria) set of decisions is required. The problem remains, however, how the compatibility functions should be derived. One suggestion is to use training data to determine this *a priori* information [92].

Markov models [8] [10] [111] [39] are used to represent the inter-class dependence (which is formulated heuristically in RL) through the use of an explicit statistical model. Markov models are used for the class or label processes in a variety of maximum *a posteriori* (MAP)

estimation methods for image restoration and segmentation [9] [39]. The local spatial interaction is modelled by a 2-dimensional Markov chain, called a Markov random field (MRF), which is expressed by

$$P(x_{\vec{\xi}_i} \mid x_{\vec{\xi}_j}, j \neq i) = P(x_{\vec{\xi}_i} \mid x_{\vec{\xi}_j}, j \in \mathcal{N}_{\vec{\xi}_i}) \tag{1.5}$$

where $\mathcal{N}_{\vec{\xi}_i}$ is the neighbourhood of $\vec{\xi}_i$. As with a Markov chain, the immediate neighbours are termed the *first order* neighbourhood, and a hierarchy of neighbourhoods can be defined by considering neighbours of neighbours (*second order*), and so on. Given the set of observations $Y = \{y_{\vec{\xi}_1}, \ldots, y_{\vec{\xi}_N}\}$, the MAP estimate of the true scene $X = \{x_{\vec{\xi}_1}, \ldots, x_{\vec{\xi}_N}\}$ is the joint labelling which maximises the *a posteriori* probability distribution $P(X \mid Y)$. From Bayes's rule [75]

$$P(X \mid Y) = \frac{p(y_{\vec{\xi}_i}, 1 \leq i \leq N \mid x_{\vec{\xi}_j}, 1 \leq j \leq N) \times P(X)}{p(y_{\vec{\xi}_i}, 1 \leq i \leq N)} \tag{1.6}$$

The denominator can be ignored during the maximisation, leaving the specification of the conditional density of the observation given the true scene and the *a priori* probabilities of the label configurations. Much of the work in this area has been concentrated in developing appropriate models for the label process, and more often than not, use MRFs to model the label interactions. Knowledge of the conditional density of the observation and the prior probabilities of the label process is either assumed, derived heuristically or parameter estimation is considered from training data or in situ (e.g. [9] [28]). The order of the MRF is often restricted to facilitate the derivation of parameters on an *ad hoc* basis [39].

One of the major problems faced by global MAP estimation is the need to optimise the estimate over the discrete space of all possible labellings. Other than in the most trivial examples the computational burden is prohibitive. For example, the optimisation problem for binary images of size 64×64 requires a search over $2^{64 \times 64} = 2^{4096}$ realisations. This has led to a number of optimisation methods that reduce the global problem to a series of

iterative, local problems, analogous to the relaxation labelling.

Perhaps the most celebrated method is that used by Geman and Geman [39] using the combinatorial optimisation procedure known as Simulated Annealing (SA) [95]. The SA procedure has its origins in statistical mechanics and was developed as a procedure for simulating the slow cooling of a solid (annealing) until it reached a low energy ground state, which also explains the physical terminology adopted when SA is used for MAP estimation. Geman and Geman show that if the image is modelled as a MRF then, under certain conditions, its probability distribution is given by a Gibbs (Boltzmann) distribution. This distribution controls the 'energy' of the system and has a control parameter, analogous to the temperature, which is gradually decreased to converge the system to ground (minimum) cost. The optimal annealing schedule is slow, however, requiring anything up to 1000 iterations over the pixel sites.

SA and MAP estimation are discussed by Besag [9], who proposed an alternative optimisation scheme called Iterated Conditional Modes (ICM). Although it only guarantees to find a local optimum, it performs well and is appealing because of its simplicity in comparison to SA and its rapid convergence (within 10 iterations). ICM is perhaps the closest parallel to the local, iterative estimation method used in this work (Chapter 3), which shares its fast convergence rate.

Multigrid methods integrate MAP estimation by MRF modelling into a multiresolution, coarse to fine analysis structure [91] [40] [73] [63]. The general principle is that at the lower resolutions of the structure, e.g. a pyramid, where there are fewer pixels, the equilibrium state is located very quickly. This estimate is then interpolated on to the next higher resolution, and is used as the starting point for the optimisation at that level. This process is repeated until full spatial resolution is achieved. In other words, the lower resolutions constrain the estimate at the higher ones. The advantages are the reduction in computational

burden, especially if a method such as ICM is used [16], and that such methods go some way to relating local and global characteristics in the image which is difficult with single resolution methods.

As already noted, multiresolution analysis allows the trade off between class and positional localisation, and there are a number of methods that exploit this. The split-and-merge techniques, developed originally by Chen and Pavlidis [24], use a linked pyramid structure and statistical decision criteria to combine global and local region information. These ideas were adopted and developed in [19] [48] [49], with variations in the linking methodologies used, but like their predecessors required prior knowledge of the size and number of regions present.

The quadtree based method of Spann and Wilson [88] made use of non-parametric classification [108] at a low spatial resolution, followed by downward boundary estimation, to regain spatial resolution. It was notable because of its efficiency and that it did not require prior knowledge of the image regions. This method was also shown to be readily extensible to the problem of texture segmentation [107] [86] [13]. Some of the successful ideas from the quadtree method have been used by Spann and Horn [87], to overcome the problems in the linked pyramid based split-and-merge method.

1.4.2 Boundary based approaches

The second class of methods for image segmentation involve the detection of the luminance transitions between regions, i.e. the boundaries (lines or edges). The fundamental importance of line and edge information in both biological and computer vision systems has long been recognised. Indeed, the biological evidence showing 'edge-detection' playing a central role in the early stages of visual perception in mammals (low level vision), such as the HVS, has often been the motivation for its adoption by workers in image processing. The biological evidence was provided by the early work of physiologists, notably Hubel and

Wiesel [52] [51] and also Blakemore and Campbell [14]. Local features, such as lines and edges, can describe the structure of a scene relatively independently of the illumination. For example, a cartoon drawing consisting only of lines is often enough for us to interpret a scene.

Image segmentation techniques based on edge detection have long been in use, since the early work of Roberts in 1965 [81]. A variety of methods of edge detection have been suggested such as: gradient detection methods, for example the Canny edge detector [22] (see also [41] [58] for others), detection of zero-crossings in the second derivative [46] [69], edge detection using edge models [53] [96], and frequency domain methods [61] [67] [20]. The bane of all these methods, however, is *noise*. Edges, by definition, are rapidly varying spatially and hence have significant components at high spatial frequencies. This is also, unfortunately, the characteristic of noise, and therefore any gradient operator that responds well to the presence of an edge will also respond well to the presence of noise or textures thus signalling *false* edges.

From an uncertainty stand point, boundary methods approach segmentation from positional localisation. There is a second manifestation of uncertainty, however, in the trade-off between spatial localisation and noise immunity. To achieve greater noise immunity requires greater spatial averaging and hence reduced spatial localisation. The extent to which the two requirements conflict is determined by the uncertainty principle of signal analysis [74] [22].

One of the ways to achieve noise immunity is by post-processing of the edge detection by some form of 'smoothing'. The work of Rosenfeld and Thurston [83], Marr [69] and Witkin [110], is notable in this respect as their approaches were aimed at combining the outputs of filters of different sizes, i.e using multiple resolutions. A robust technique for edge detection and orientation estimation, based on a multiresolution signal model was introduced in [102], and is considered in more detail in Chapter 4.

Having obtained an edge map, there is usually a second stage to boundary based segmentation, which is to group the boundary elements to form lines or curves. This is necessary because, other than in the simplest, noise free images, the edge detection will result in a set of fragmented edge elements. There are three main ways that this has been attempted: local linking techniques (see [5]), global methods, such as Hough transform (HT) methods [34] [4], or combined approaches, such as the hierarchical HT [80] and the MFT based methods [20] [30].

The local linking methods use attributes of the edge elements, such as magnitude, orientation and proximity, to grow the curves in the image. Heuristic, graph-theoretic and relaxation methods [65] have been employed to achieve the line linking. In the HT methods (see [56] for a review), the edge elements are transformed to a parameter space, which is a joint histogram of the parameters of the model of line or curve being detected. The peaks in this histogram then indicate the presence and location of the lines or curves being detected. Recently a 'hierarchical' HT [80] has been developed to make the method more robust and efficient, by first doing HTs in small windows across the image (8×8) and then reapplying the HT technique recursively to group together the detected line segments into longer and longer lines. The curve extraction process implemented using the MFT [20], is notable as it performs both the edge detection and curve extraction in a single framework.

1.4.3 Combined methods

The third category of methods are those that have implicitly or explicitly combined a region based approach with edge detection, and are of special interest to the work presented here. Regions and their boundaries are complementary, so it seems appropriate to use a cooperative scheme to overcome the uncertainty encountered in either scheme alone. From the above discussions about uncertainty, in a cooperative scheme the image segmentation is tackled simultaneously from high class localisation (regions) and high positional localisation

(boundaries).

In the MRF methods the edge detection is incorporated by the introduction of a binary valued line process, also modelled using MRFs [39] [38] [43]. The line process modelling described by Geman and Geman [39] is representative of the approach used. The pixel sites of the region process are augmented by line sites which are placed midway between each vertical and horizontal pair of pixels. The configuration space of any neighbourhood is thus extended to include all possible pairs of allowable grey level and line states. The energy function, which determines the prior from the current neighbourhood configuration, is decomposed into contributions from both the current grey level state and line state (which itself prohibits certain pixel configurations). Although the results are better with the introduction of such a line process, the problem of parameter estimation for the joint model is compounded. Also, the size of the line process neighbourhood is artificially tied to the grey level process, which itself is kept small to make the computation tractable.

The combining of a region based segmentation with edge detection in a more explicit manner is not new. Some region 'growing' methods have used edge operators to prevent the merging across boundaries (see review by Harlick [47]). The use of a more general, multiresolution and cooperative segmentation is described by Hanson and Riseman [45]. Recently Pavlidis and Liow [77] (also [15]), have taken the results of a split-and-merge, linked pyramid approach and added a boundary relaxation step to refine and improve the region errors and enhance the resultant contours. By incorporating global shape and shadow information they have also shown good results in restricted applications, such as extracting buildings from aerial images [66].

1.5 Requirements of a Segmentation Approach

The above discussion highlights a number of problems which the work presented in this thesis seeks to address. These are:

1. Prior Knowledge. There are two levels of this: first, there is the fundamental problem of any model-based approach, that of parameter estimation and second, to what degree and in what form should higher level knowledge be incorporated into the segmentation process.

   Part of the weakness of the MRF methods is that parameters of the conditional distribution governing the observation process have to be estimated, often from training data. Also the parameters of the prior distributions have to be determined, often by empirical or *ad-hoc* means. Ideally, the segmentation would be unsupervised, with parameter estimation performed in situ.

   The point about the higher level knowledge is perhaps in the realms of psychophysics: to what extent does visual knowledge about what we know influence what we see? It would be desirable to make minimal assumptions, keeping the segmentation process mostly at a low level.

2. Local Processing. This is an obvious consequence of the intrinsic structure of images, i.e. the features of interest to any segmentation are local and that the size of these localities varies, from feature to feature, across the image. This has profound implications for the model underlying the segmentation.

   An additional benefit of local processing is that parallel processing can be used in computation.

3. Uncertainty. It must be possible to obtain sufficient resolution in both class and position to answer the 'what is where' question [104]. Also there is a need for noise

immunity when performing edge detection. Both of these suggest a multiresolution approach.

4. Computational Efficiency. The method should be computable and not make unreasonable demands on storage and CPU time. This depends partly on the representational structures used, and should result in a segmentation which is in some sense a 'minimal' representation of the image.

5. Flexibility of Image Model. The image model should be extensible to perform a generalised segmentation. The obvious extension is from regions of homogeneous grey level to regions of homogeneous texture.

1.6 Thesis Outline

In Chapter 2 a suitable model for images on which a segmentation can be based is derived from a general class of linear multiresolution image models [101] [25]. It is shown how a multiresolution model can succinctly describe image properties which span a number of scales, such as regions and boundaries. A segmentation algorithm is introduced which assumes the described model. The algorithm handles region and boundary information separately and cooperation is achieved through an explicit interaction process. The key aspects of the method are: scale selection, adjacency graphs and iterative estimation.

The region estimation process of the segmentation is detailed in Chapter 3. The construction and properties of low-pass image pyramids, parameter estimation and scale selection are considered. A new iterative, decision-directed estimation method is then described, which operates on a region adjacency graph representation of the image. Experimental results are presented that illustrate its operation.

Chapter 4 describes the boundary estimation part of the segmentation. A robust orientation estimation technique is presented and it is shown that by combining orientation

estimates from a number of levels of a pyramid of the input image, it is possible to considerably reduce the estimation errors. The performance of the algorithm is demonstrated for input images over a range of SNRs.

A scheme for boundary segmentation which operates on an orientation pyramid of the image is defined. The resulting boundary representation is a 'dual' of the region adjacency graph of the region based process. Again the processing involves the three stages of scale selection, boundary graph creation and iterative estimation. Results are presented that illustrate the operation of the boundary processing.

In Chapter 5 the two estimation strategies of Chapter 3 and 4 are brought together and an interaction process defined for the combined segmentation. Results of running the combined segmentation method on a number of synthetic and natural images are presented and its performance is evaluated.

The thesis is concluded in Chapter 6 with a synopsis and a discussion of further work.

Experimental and Display Conditions

All test images are 256×256 pixels in size and eight bit (0 to 255). White Gaussian noise was generated by the polar method [60] using the nominally uniformly-distributed ouput from the UNIX[1] random number generator *random()*.

All algorithms were coded in the C++ programming language [89] on SUN 3/160 and SUN 4 SPARC work stations running UNIX v4.2. The results were photographed with a Dunn Instrumentrs Multicolour unit, using Kodak TMAX 100 ASA monochrome film on the green channel with a nominal exposure time of 2.36 seconds, and Fuji 100 ASA colour film with exposure times of Red: 2.85 seconds, Green: 2.36 seconds and Blue: 0.83 seconds.

[1] UNIX is a trade mark of AT&T Bell Laboratories.

Chapter 2

Towards a Model for Segmentation

2.1 Introduction

This chapter is concerned with an appropriate model for images on which a segmentation can be based. The proposed model is derived from a general class of linear multiresolution image models [26] [25] [103], forms of which have been successfully used in the areas of image estimation [27], coding [93] and curve extraction [20]. It will be shown how the basic model may be generalised to model both region and boundary features and how the two feature descriptions can be made to interact within the same framework.

A mathematical definition of the model is given and an appropriate adaptation introduced that generalises the representation of local features in the image to include both regions and boundaries. The chapter is concluded with the resulting forms of the estimation and segmentation algorithms, based on the model.

2.2 Multiresolution Image Models

The modelling of images using multiresolution representations is not new - there are numerous examples of the use of multiresolution representations such as quadtrees and pyramids in the segmentation of images (e.g [24] [19] [88]). It is only recently, however, that the whole

question of the multiresolution image modelling has been analysed and the seemingly disparate models been put in to a unified framework. The work of Clippingdale [25] described a new class of linear multiresolution image models and an efficient quadtree estimator based on such models. More recently the paper by Wilson *et al* [103] has attempted to identify the similarities, differences and properties of the varieties of multiresolution image models that have been used in estimation, coding and segmentation.

The major motivation behind the use of multiresolution models is the ability of such models to describe image properties that span a range of scales, which directly reflects the nature of most images. For example, typical images consist of regions of varying size with homogeneous properties such as luminance, colour and texture, separated by more or less sharp boundaries [109]. This can be succinctly modelled by use of multiple resolutions. Small features and fine detail are modelled at relatively high spatial resolutions, while large features are modelled at low spatial resolutions. The image can thus be visualised as being built up by successive approximation through scale, beginning with a coarse description at a low resolution and progressively refining this until the image plane is reached. Such models have the attraction that they result in estimation procedures which can be computationally efficient, requiring only about 30% more storage for a pyramid than the original image, and giving effective processing window sizes from 2×2 to the whole image.

2.3 Linear Multiresolution Image Models

The estimation, coding and curve extraction methods described in [25], [21] and [94] all use varieties of a general class of linear multiresolution image models first described in [26]. Before giving a mathematical definition of the general model and describing an appropriate variation for the segmentation task in hand, it is perhaps worth describing the example used by Calway [21] for modelling curves. This will serve to reinforce the main concept of

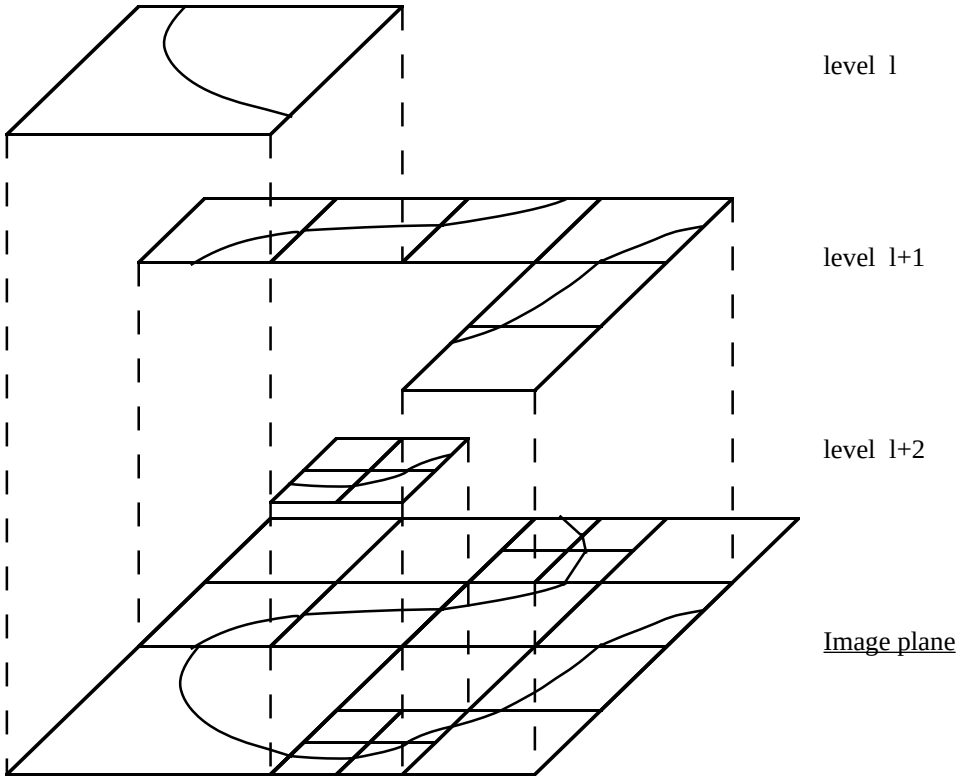


Figure 2.1: A multiresolution image model for curves

the modelling of features over scale.

Figure 2.1 shows a typical realisation of a model for curves. Local image features are represented at different scales in the model, creating an inhomogeneous *tessellation* of the image into square regions (blocks) of different sizes. The larger features are represented by larger blocks from lower spatial resolutions whereas detail, e.g. where there is high curvature, is represented by smaller blocks. There is a constraint imposed on these feature regions so that there is only a *single* local feature within each. The main point to note about this image model is that a quadtree sub-division of the image is employed, which leads to a simple structure and has computational advantages when deriving suitable estimation schemes. It is also not hard to see that even if the feature *prototype* were different, say a region property such as grey scale, rather than edges or lines, then a similar type of

structure could be used. Such a variation is in fact the basis of the estimation techniques described in [26] [25].

2.3.1 Mathematical Definition

The above image model may be defined by the following recursive operation

$$s_{ij}(l) = \sum_{mn} A_{ijmn}(l) s_{mn}(l-1) + \sum_{pq} B_{ijpq}(l) w_{pq}(l) \tag{2.1}$$

where $s_{ij}(l)$ is the 'image' at level l of the model and is given by taking a *linear* combination of the level above plus an *innovations* image $w_{pq}(l)$. In this case the limits of the summation indices are $0 \leq m, n, p, q < 2^{M-l}$, which keeps the successive image levels the *same* size. The ranges of m, n, p and q would differ if the model represents levels of images of different sizes as in the quadtree or pyramid models in [26]. The model is then parameterised by linear operators, which 'construct' the image by controlling the features of the parent level and the innovations used to form the child level. The curve model in figure 2.1 is particularly simple to generate as these linear operators act as selection functions taking an appropriate quadrant of the previous level or simply the innovation level. The initial conditions are

$$s_{ij}(0) = \sum_{pq} B_{ijpq}(0) w_{pq}(0) \tag{2.2}$$

and the resulting image $s_{ij}(M)$ is simply given by level M of the model [25] [93].

2.4 A Generalised Image Model

Nothing explicit has yet been said about the form of the feature innovations. In the case of the curve model, each block is a locally defined real function which has an associated orientation and position vector, which models the single boundary feature in each block [20]. As mentioned above, however, the feature prototype may be a different local image attribute such as grey level. In this work, the objective is to generalise the model to include both

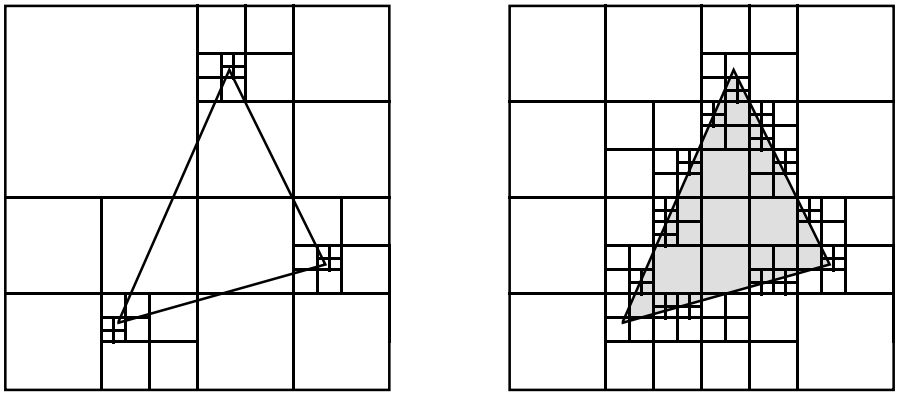


Figure 2.2: Model realisations for boundary and region features

boundary feature types, with the need to model corner features and junction points, and region feature types. There are two aspects to this generalisation of the model [20]:

1. The definition of the feature innovation $w_{pq}(l)$ of equation (2.1) should include both boundary and region feature types.

2. It should allow each spatial region to have many features defined at a *number* of different scales.

The first condition is obvious, but the need to suitably model linear features, corners and branch points is more problematic. It is convenient to modify the structure shown in figure 2.1 so that, rather than representing the region boundary as a series of polygon sides, the vertices of this polyline form the representation. This modification has the advantage that corners features now have an explicit representation and if these vertex nodes are allowed to have degree greater than 2 then it is possible to represent junction points.

The second generalisation is made clearer by considering the example in figure 2.2 which shows the realisations of the multiresolution model for boundary and region feature prototypes separately. The region model displays the characteristic block tessellation where again under a single feature per block constraint, the region 'interiors' are larger spatial

regions and are represented at the lower spatial resolutions. It is important to note that the resulting tessellations are different. There are, however, similarities in the structure: where there is local *variation* in the feature prototype, i.e. high curvature in the case of boundaries, or edges between different regions, this results in smaller blocks, while slow variation is represented by larger blocks. Indeed, this is the successive approximation nature of the general class of multiresolution models under consideration.

For the above second condition to be met, each spatial region is allowed to have both a local boundary and region attribute. So for every spatial block of the boundary model, there is an associated number of region blocks and vice versa.

2.4.1 A Model for Segmentation

There are two main parts to the generalised model used in this work: a boundary process which uses a vertex based representation to generate a set of connected curves in the image; and a region process which models region interiors as an MRF [8] [39], where the adjacency of the pixels is constrained by the boundaries within the boundary process.

1. Boundary Process

A set of vertex points, $X(l)$ is defined to represent the image boundaries that have the spatial co-ordinates $\vec{\chi}_i(l), 1 \leq i \leq N(l)$ at scales $M \geq l \geq 0$. A set of connections $L(l)$ is defined between pairs of vertex points

$$L(l) = \{\lambda_{ij}(l),\ 1 \leq i, j \leq N(l), /; i \neq j\} \tag{2.3}$$

where

$$\lambda_{ij}(l) = 1 \text{ if } \vec{\chi}_i(l) \text{ is linked to } \vec{\chi}_j(l) \tag{2.4}$$

for $i \neq j$. This creates a graph of the boundaries at each scale $G(l) = \{X(l), L(l)\}$.

An initialising graph $G(l_0)$ is defined at some level $l_0 > 0$. The evolution of the vertex

points is modelled through scale by using the conditional probabilities

$$P(\vec{\chi}_{ik}(l+1) \mid \vec{\chi}_i(l), L_i(l)), \quad 0 \leq k \leq 3 \tag{2.5}$$

subject to the initial conditions in the form of a distribution $P(\vec{\chi}_i(l_0))$ at the largest scale and where $L_i(l)$ is the set of links associated with $\vec{\chi}_i(l)$. Each vertex point therefore has its position refined over successive scales and may be split into up to 4 new points within a quadtree structure.

The children of $\vec{\chi}_i(l)$ are themselves restricted to form a connected sub-set. If the vertex 'splitting' is expressed as a mapping

$$\vec{\chi}_i(l) \rightarrow X_i(l+1) = \{\vec{\chi}_{i0}, \ldots, \vec{\chi}_{ik}, 0 \leq k \leq 3\} \tag{2.6}$$

then the connectedness is maintained by creating the links

$$\lambda_{ik,ik+1}(l+1) = 1, \quad 0 \leq k < \text{card}\{X_i\} \tag{2.7}$$

Also the ordering of the conditioning set $X_i(l)$ is maintained in the child set, even with the addition of new points $\vec{\chi}_{ik}$, by creating links from the new points to the existing ones. For example, if $\lambda_{ij}(l) = 1$ then $\lambda_{in,j}(l+1) = 1$, where $n = \text{card}\{X_i(l+1)\}$. Figure 2.3 illustrates the process over 4 levels.

The conditional probabilities $P(\vec{\chi}_{ik}(l+1) \mid \vec{\chi}_i(l), L_i(l))$ can be regarded as configuration probabilities in the sense that they govern the shape of the local boundary at the child level, conditioned on the shape at the vertex point on the parent level.

Although the above does not allow for the creation of new 'objects', this could be incorporated into the process by having a scale dependent probability for the formation of new points, in addition to those from the initialising set. These probabilities would, in general, decrease at the smaller scales reducing the chance of new regions being created as the image plane is reached.

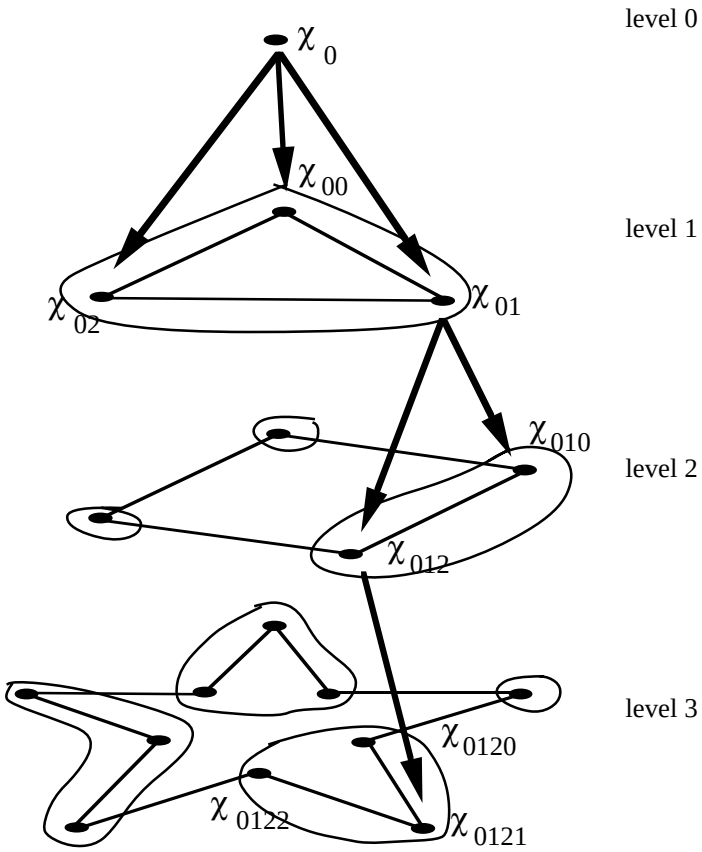


Figure 2.3: Example of evolving boundary

2. Region Process

The second stage of the model is the generation of the regions within the boundaries defined by the graph defined at the image level $G(M)$. Once more, the quadtree structure is used, in which the spatial co-ordinates $\vec{\xi}_i(l)$ represent square regions of the image. Each $\vec{\xi}_i(l)$ is associated with a unique region $\Lambda_i(l)$ of the image

$$\Lambda_i(l) = \bigcup_{\substack{2^{M-l}\xi_{i0} \le p \le 2^{M-l}(\xi_{i0}+1) \\ 2^{M-l}\xi_{i1} \le q \le 2^{M-l}(\xi_{i1}+1)}} (p,q) \tag{2.8}$$

These sets are clearly disjoint, $\Lambda_i(l) \cap \Lambda_j(l) = \emptyset$. A coarse-to-fine refinement strategy is used to split each region block into 4 if the block contains either a vertex point $\vec{\chi}_j(M)$ or

an intersecting boundary link $\lambda_{rs}(M)$. Expressed as conditional probabilities

$$\begin{aligned} P(\vec{\xi}_{ik}(l+1) \mid \vec{\xi}_i) &= 1, \text{ if } \vec{\chi}_j(M) \in \Lambda_i(l) \text{ or} \\ &\quad \lambda_{rs}(M) \cap \Lambda_i(l) \neq \emptyset \\ &= 0 \text{ else} \end{aligned} \tag{2.9}$$

where $\vec{\xi}_{ik}$ are the four children of $\vec{\xi}_i$ and $0 \leq k \leq 3$, for some j, r, s. If there are no boundary features in a given region block then it is classed as being interior and the quadtree is truncated at that point.

This process results in a unique tessellation of the image into a set of square blocks where any part of the image is modelled by the largest square block that will fit within the boundary. Figure 2.4 show the tessellation process over several levels. At any level of the model there are two classes of pixels: those that are interior region pixels and those containing a boundary segment. This process can be seen to continue past the image plane to a sub-pixel level, with the result at the image plane being the best level of approximation possible in the model.

The signal process operates within the above selection process. Each node $\vec{\xi}_i(l)$ has the associated datum $x_{\vec{\xi}_i}(l)$, or simply $x_i(l)$. The values are calculated as a linear combination of the neighbours of $\vec{\xi}_i(l)$, denoted by the set $\mathcal{N}_i(l)$, plus a term drawn from a zero mean, Gaussian white noise process

$$x_i(l) = \sum_{j \in \mathcal{N}_i(l)} \alpha_{ij} x_j(l_j) + \beta_i w_i(l) \tag{2.10}$$

The signal model is in fact a Gaussian MRF [111] [23] [85] [37] defined on the irregular lattice structure defined by the region tessellation $\vec{\xi}_i(l)$ and thus possesses the 2-dimensional Markov property [8] [9] [39]

$$P(x_i \mid x_j, j \neq i) = P(x_i \mid x_j, j \in \mathcal{N}_i(l)) \tag{2.11}$$

which expresses the dependence of the value at the pixel $\vec{\xi}_i$ on the neighbourhood $\mathcal{N}_i(l)$.

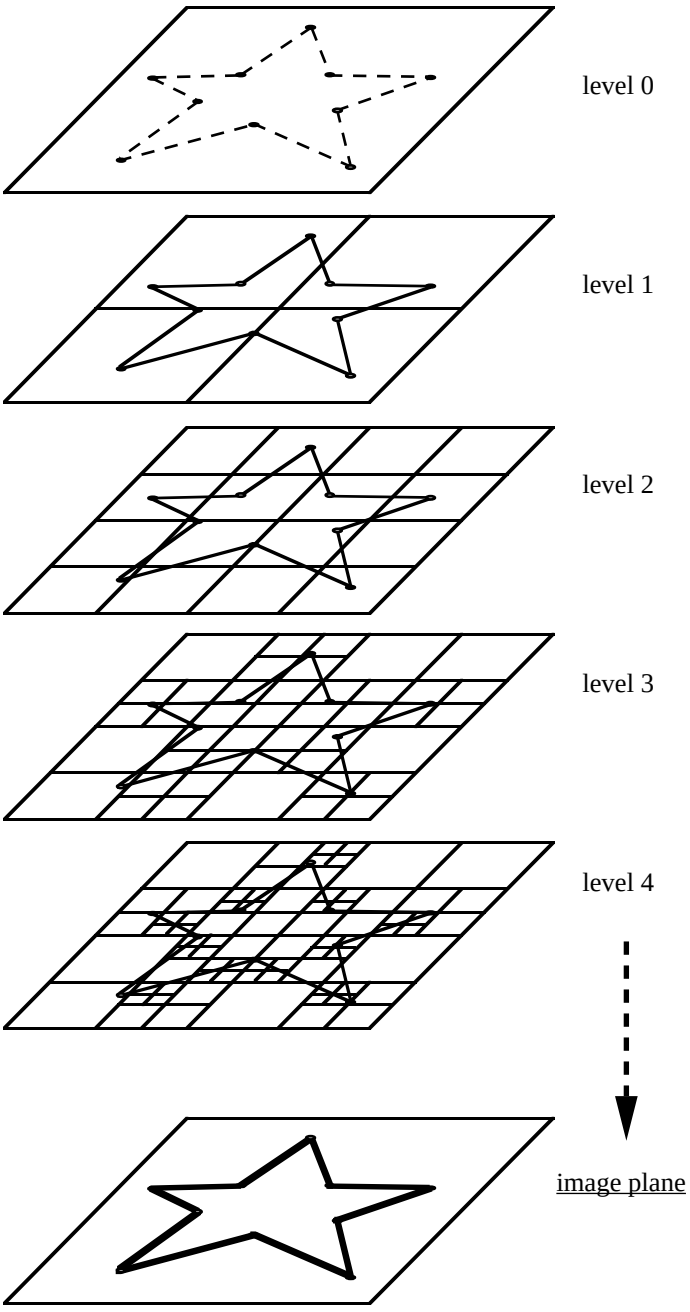


Figure 2.4: Region tessellation

2.5 A Segmentation Algorithm

The purpose of this section is to outline the segmentation process based on the multiresolution image model described above. The use of a multiresolution structure generally involves at least a two-pass algorithm: the first pass is the building of the structure; and the second performs the required processing. The multiresolution structure which serves as the basis of the segmentation algorithm in this work is a low pass pyramid representation. Low pass pyramids, of which the quadtree is an example [90], are fast and easy to generate if a small size, Gaussian approximation, kernel is used [18] [70] [99]. The generation and properties of pyramids are considered in more detail in Chapter 3 and in Appendix B.

The segmentation then consists of three processes that operate on the pyramid. The overall structure is depicted in figure 2.5. The region process selects nodes from the pyramid and builds a modified form of a region adjacency graph [64] [72]. At the same time the boundary process uses orientation estimation on the pyramid to generate a boundary graph. Each process then refines these initial estimates using iterative 'Bayesian' estimation within the neighbourhoods defined by the adjacency graphs. The interaction process arbitrates between the two otherwise autonomous processes, allowing the exchange of information through the interaction neighbourhoods defined by the model. The region and boundary processes have a common structure which may be summarised as

1. Node selection: A set of nodes are selected from the appropriate pyramid representation using maximum likelihood (ML) scale selection to form an initial segmentation.

2. Adjacency graph: These nodes are linked together to form an adjacency graph.

3. Iterative estimation: Each site of the adjacency graph is updated conditioned on the values of its neighbours, plus any interaction from the cooperating process.

   The overall paradigm is essentially to produce a 'least-cost' or minimum mean square

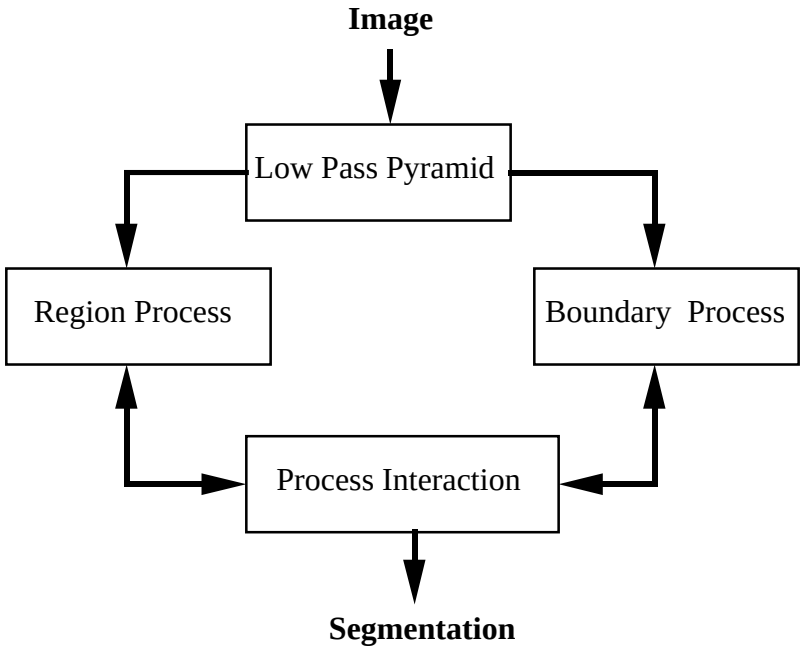


Figure 2.5: Overall algorithm structure

error (MMSE) fit to the data, within the constraint of the image model.

1. Node Selection

The main justification behind using a multiresolution model is that the representation of image features is simplified through the use of window scales that reflect the sizes of image features. The aim of the node selection is to tessellate the image into the smallest set of disjoint regions for which the data are consistent with the model. In other words, in each region, the *largest* block for which the data are compatible with the model is sought. There is an immediate data reduction by this initial segmentation, which has computational benefits for the subsequent iterative processing. Two initial segmentations are created, one for each of the two branches of the overall processing shown in figure 2.5.

The tessellation which results from this node selection step overcomes the problems of context which are associated with single resolution methods. The context of each node of the initial segmentation is taken across a number of different scales, thus both local and

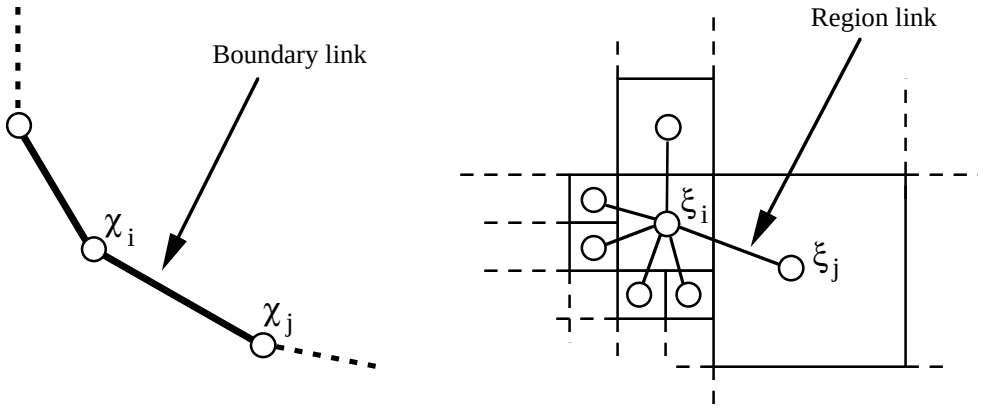


Figure 2.6: Boundary and region adjacency neighbourhoods

global interaction is represented in this structure. Furthermore, the compression of the data from, say, $256 \times 256 = 65536$ pixels to typically 500 or so nodes (Chapter 3) considerably reduces the computational burden. Both of these advantages are the justification of the use of multigrid methods in MAP estimation (see review in Chapter 1).

2. Adjacency Graphs

Boundaries in the image form an adjacency graph in which each node has a neighbourhood defined as those nodes to which it is connected by boundary segments. The region model can also be seen as an adjacency graph, where each spatial region can be represented by a node and all the surrounding spatial regions are its neighbours (figure 2.6). In the boundary graph, the value at the node is its position, which is free to vary, conditioned on the values at the neighbours. In the region graph, the value at the node is some homogeneous region property such as grey level, again conditioned by a *first order* neighbourhood.

There is no strict duality between the above two representations, in other words it is not possible to replace say the 'regions' of the boundary graph by nodes to get the region adjacency graph. It is, however, possible to define an interaction neighbourhood such that each entity in either graph can be related to its 'dual' entity in the other graph. The

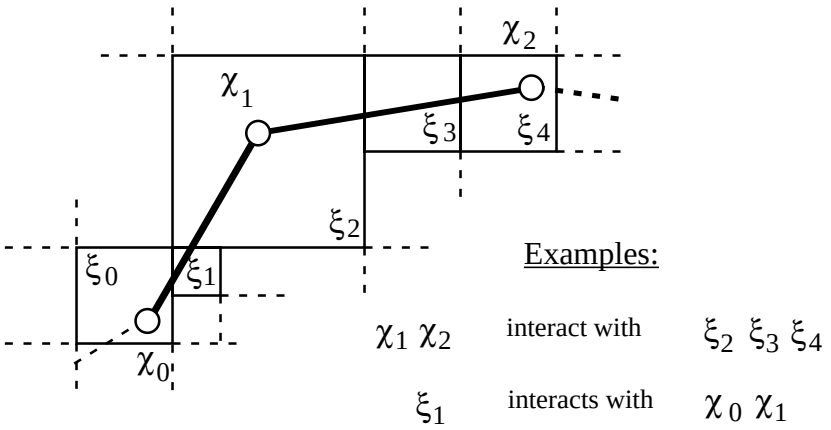


Figure 2.7: Interaction neighbourhood

simple approach adopted in this work is for each boundary link to be associated with all region blocks it intersects and each region block to be associated with all the links that pass through it, and hence the boundary nodes at either ends of these links (figure 2.7).

As already noted, each graph is able to provide the appropriate context for the features being represented. By explicitly separating the region and boundary process structures, the size of the context of one is not artificially tied to that of the other, which is a weakness of combined MRF region and line process models. So, for example, 'global' boundary information is able to influence relatively localised region link decisions.

3. Iterative Estimation

The iterative processing updates the initial segmentation obtained by the multiresolution inference processes using a combination of a MAP decision approach and ML estimation. The decision processes using the structure of the adjacency graphs are binary and based, more or less, on the following MAP decision rule

$$\text{Accept } H_0 \text{ if } P(H_0 \mid \text{data}) > P(H_1 \mid \text{data}) \tag{2.12}$$

which from Bayes's rule gives

$$\text{Accept } H_0 \text{ if } p(\text{data} \mid H_0)P(H_0) > p(\text{data} \mid H_1)P(H_1) \tag{2.13}$$

where H_0 and H_1 are the alternative hypotheses being tested. Where there is no knowledge of the prior probabilities, $P(H_0)$ and $P(H_1)$, this test can be simplified to the ML by assuming that $P(H_i) = 0.5$.

The MAP decision rule of equation (2.12) is the minimum 'probability of error' decision rule and is appropriate in the absence of an application specific cost [92] [33]. Thus equation (2.12) is the minimum Bayes's *risk* decision rule assuming a symmetrical cost function, i.e. it weights errors equally [36]. The two main stages where such a test is used are in the link decisions for the neighbourhoods of the adjacency graphs and during the block splitting, both described in Chapter 3.

The general paradigm of the estimation procedures is that of MMSE estimation which in the case of Gaussian data is equivalent to a ML estimation [71]. MMSE or 'least-cost' methods are the basis of the orientation estimation and the boundary refinement (Chapter 4), where the 'best' fit to the data is sought within the constraint of the region and boundary model that has been outlined in this chapter.

All parameter estimation is performed in situ. Thus the algorithm is unsupervised and adaptive to the data. This is important if the processing is to be kept as general as possible. Statistics are gathered locally within the multiresolution structure, thus allowing local image variation.

Another aspect of keeping a segmentation general purpose is the form of the prior knowledge used. As noted in the discussion in Chapter 1, ideally minimal assumptions should be made. In this work the prior knowledge is incorporated into the image model where the main assumptions are: that regions are compact and homogeneous, under some property, and that boundaries are sharp and more or less connected.

2.6 Summary

This chapter has proposed a framework for segmentation. It has been shown that by extending the general class of linear multiresolution models to include multiple feature prototypes and lifting the restriction on the number of scales for the representation of a given image locality, it is possible to derive an image model for both boundaries and regions. Also by using modified adjacency graphs, lateral processing can be incorporated and an arbitration process set up to achieve an unambiguous segmentation.

The next three chapters describe the region, boundary and interaction processes. The results of experiments are presented and discussed at the end of Chapter 5.

Chapter 3

Region Estimation Process

3.1 Introduction

This chapter describes in detail the region estimation process within the image model and segmentation framework outlined in the previous chapter. The region process has three main components that operate on a low-pass pyramid representation of the image: first there is a node selection process which gives an initial estimate of the region classes, next neighbourhoods are defined for these nodes and a modified form of a Region Adjacency Graph (RAG) formed and finally an iterative estimation method, employing a MAP decision process, is used to group together the nodes to form homogeneous regions.

3.2 Lowpass Pyramid

The input to the region estimation process (and boundary process) is a lowpass *pyramid* of the image to be segmented (figure 2.5). As described in Chapters 1 and 2, pyramid image representations have been used successfully by several image segmentation schemes, such as [24] [19] [88] [87] . Pyramids afford fast and simple smoothing of an image providing a set of images, decreasing in resolution, each with a progressively greater signal to noise ratio (SNR). The general form of the processing is a combined lowpass filtering, using a so called generating kernel, and sub-sampling or decimation by a constant factor. It is common to

sub-sample by a factor of 2, which is both simple and efficient to implement. The successive smaller levels can be visualised as being stacked one on top of the other with the original image at the base, creating a square-based pyramid structure.

Given an image $x_{ij}, 0 \leq i, j < N = 2^M$, and a lowpass kernel A_{mn}, the general form of the processing is the following recursive operation:

$$x_{ij}(l) = \sum_{m=-K}^{K} \sum_{n=-K}^{K} A_{mn} x_{(2i-m)(2j-n)}(l+1) \tag{3.1}$$

where there are $M+1$ levels, $0 \leq l \leq M$, and the base of the pyramid is the image $x_{ij}(M) = x_{ij}$ and the kernel is of size $(2K+1) \times (2K+1)$. Note that the scale index l appears as the argument of the function $x(l)$, whereas the spatial indices are given as the subscripts x_{ij}.

The simplest form for the generating kernel is a two by two kernel with coefficients $A_{mn} = 0.25, 0 \leq m, n \leq 1$ which forms a *quadtree* [90]. It can be easily verified that this rather simple block averaging causes aliasing of spatial frequencies above $\pi/2$ radians [86]. As it was also the purpose to use the same pyramid for the *boundary* process this would have also greatly biased any orientation estimation. Small radius kernels that have more nearly an ideal frequency response, such as the Gaussian approximation kernels of Burt and Adelson [18] or [70] were considered, but found not to be circular symmetric, which is again desirable for orientation estimation. Appendix B describes a new design strategy for the design of small filter kernels with specified radial and angular frequency response. This was used to generate the lowpass (and edge) kernels used in the present work. A full account of the performance of these new kernels is given in [102], where it is shown to be significantly better than the commonly used approximation to a Gaussian response. Table 3.1 lists the coefficients of the 4×4 lowpass kernel chosen for the present work. The main reason for using this rather than the better 5×5 kernel (see Appendix B) is that a quadtree decimation can be employed, which is generally the case for *even* sized kernels.

.0109	.0582	.0582	.0109
.0582	.1227	.1227	.0582
.0582	.1227	.1227	.0582
.0109	.0582	.0582	.0109

Table 3.1: Optimised lowpass filter kernel size (4×4)

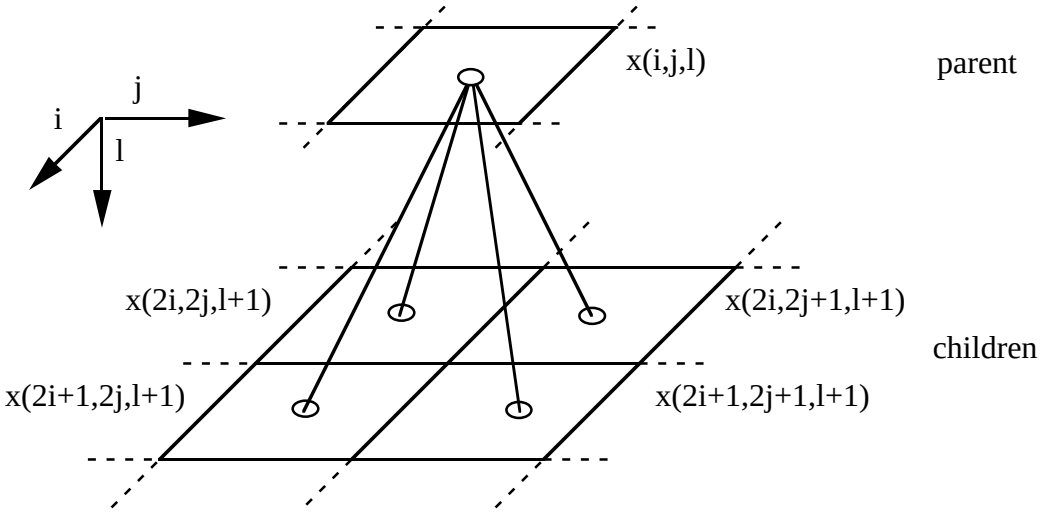


Figure 3.1: Quad-tree structure

A similar type of argument favours the use of *odd* sized kernels for orientation detection, where the estimate becomes centred on the pixel.

Although the quadtree kernel was not used for the filtering, the quadtree structure is used in the region processing, where non-overlapping groups of four nodes (*children*) at a particular level, notionally have a single *parent* at the level above (figure 3.1). The single node at the lowest resolution is then the root of the tree.

3.3 Region Node Selection

It has already been noted how a lowpass pyramid trades off spatial resolution for noise reduction, so it then becomes possible to use the gain in signal to noise to make, at some level, a reliable decision on whether a node is an interior or edge node. Under the model of the image given in Chapter 2, there is a pyramid node at a lowest resolution which

represents the underlying, square image region. The parent of this node is then an edge node, having children in more than 1 region. At this point it is to be expected that the inter-regional signal variance exceeds the 'within' region variance. The problem is then to determine a good estimate of the necessary statistics from which a reliable scale selection decision can be made.

3.3.1 Local Parameter Estimation

A suitable estimator for the 'within' region variance of a node can be derived from the expected value of the squared error between children and parent values [25] (figure 3.1). Using integer division to calculate pixel indices:

$$\begin{aligned} v^2(l) &= E[(x_{ij}(l) - x_{i/2,j/2}(l-1))^2] \\ &= E[x_{ij}^2(l)] - 2E[x_{ij}(l)x_{i/2,j/2}(l-1)] + E[x_{i/2,j/2}^2(l-1)] \end{aligned} \tag{3.2}$$

given that $E[x^2(l)] = \mu^2 + \sigma^2(l)$, which corresponds to a constant signal within the region and additive white noise. By assuming a signal to noise gain of 4, which is the case if a quadtree is used and simplifies the analysis, then $E[x^2(l-1)] = \mu^2 + \frac{1}{4}\sigma^2(l)$ and

$$\begin{aligned} v^2(l) &= \mu^2 + \sigma^2(l) - 2E[x_{ij}(l)x_{i/2,j/2}(l-1)] + \mu^2 + \tfrac{1}{4}\sigma^2(l) \\ &= \sigma^2(l) - \tfrac{2.\sigma^2(l)}{4} + \tfrac{1}{4}\sigma^2(l) \\ &= \tfrac{3}{4}\sigma^2(l) \\ \Rightarrow \sigma^2(l) &= \tfrac{4}{3}v^2(l) \end{aligned} \tag{3.3}$$

The middle covariance term reduces by noting that in construction of the pyramid, each parent $x_{i/2,j/2}(l-1)$ is itself a linear combination of the children $x_{ij}(l)$.

Table 3.2 shows how the inter-level variances compare with the true signal $E[x^2(l)]$ and noise variances $\sigma^2(l)$ for the synthetic 'shapes' image which has a SNR = 0dB (figure 1.1(a)). At the lower levels the inter-level variance corresponds well with the actual noise as the signal variance is dominated by the noise. At higher levels of the pyramid, however, this statistic is mainly influenced by the signal variance, as the noise variance at these levels is close to zero.

level l	$E[x^2(l)]$	$\sigma^2(l)$	$v^2(l)$
8	459.8	399.0	450.2
7	93.0	35.9	33.3
6	61.4	9.1	10.7
5	49.6	4.9	9.1
4	36.9	4.6	11.4
3	18.4	4.3	19.8
2	3.5	3.6	2.6
1	0.7	3.2	1.7
0	0.0	4.0	0.0

Table 3.2: Inter-Level noise variance estimates on 'shapes' SNR = 0dB

For the node selection process a local average value $\overline{v}^2_{ij}(l)$ for the squared error is calculated by taking the average squared error in a region around the node (i, j) being tested

$$\overline{v}^2_{ij}(l) = \sum_{m=i-R}^{i+R} \sum_{n=j-R}^{j+R} v^2_{mn}(l) \tag{3.4}$$

This is can be done efficiently by pyramid smoothing of the squared errors, giving local block averages. The blocking artifacts introduced do not adversely affect the final node selection. The extent of the locality, determined by R, of the expected value can be varied by taking estimates from higher levels of the averaging pyramid. The process is illustrated in figure 3.2.

This process is a form of *hierarchical normalisation* [20], where the squared error to be used in the node selection test is normalised by the local average of the same error. It is aimed at emphasising variability which is significant in the local context, and is consistent with the model of features being local in extent. A more direct form of this normalisation process is used in the orientation estimation (see Chapter 4).

3.3.2 Selection Procedure

The node selection procedure is performed bottom-up on the grey level pyramid using a quadtree structure. Each group of 4 children is tested to see if they are interior to a region, or an edge runs through them, by comparing the average squared error between them and

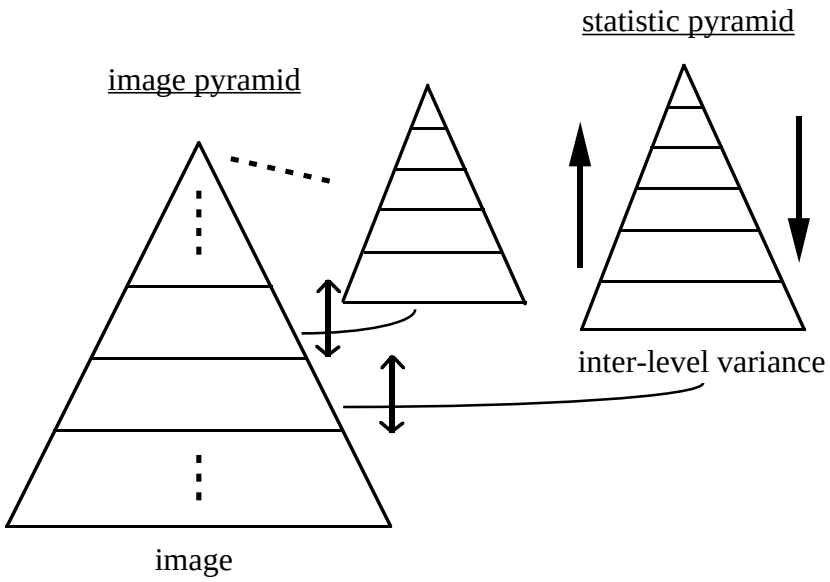


Figure 3.2: Local parameter estimation

their parent against the local 'expectation' of this squared error.

It was found that the best selection was obtained by comparing the block average with an appropriate fraction of the local variance estimated for the level below. This may be due to the fact that at higher levels the variance is influenced largely by the signal variance. The impact of this is that the lowest possible level at which a selection can be made is level $M-1$, one above the image plane. This restriction is not unreasonable, since with images where any significant noise is present it is unlikely that a reliable decision at the pixel level can be made. For images where there is little or no noise present, this is not a problem either, as the lost spatial resolution would be recovered at the region refinement stage of section 3.5.2.

The scheme can be summarised as:

1. For levels $l < M$, test the average child-parent squared error against the local average from the level *below* ($\overline{v}^2(l+1)$). If

$$\frac{1}{4}\sum_{0\leq m,n\leq 1}\frac{4}{3}(x_{i+m,j+n}(l)-x_{i/2,j/2}(l-1))^2 > \alpha\overline{v}^2_{2i,2j}(l+1), l<M \qquad (3.5)$$

then mark the child nodes. The extent of the accepted variability is controlled by α.

2. Select the lowest marked nodes, i.e those at the highest resolution. Once a node is marked for selection the quadtree is terminated at this point, saving any further testing.

If the selected node $x_{ij}(l)$ is denoted by the scale-space position vector $\vec{\xi} = (i, j, l)^T$ and the n selected nodes represented by the set $R = \{\vec{\xi}_0, \vec{\xi}_1, \ldots, \vec{\xi}_{n-1}\}$, then each node $\vec{\xi}$ represents the disjoint set of image pixels $\Lambda_{\vec{\xi}}$

$$\Lambda_{\vec{\xi}} = \bigcup_{0 \leq m,n < 2^{M-l}} (2^{M-l}i + m, 2^{M-l}j + n, M) \tag{3.6}$$

where the blocks are non-overlapping such that

$$\Lambda_{\vec{\xi}_i} \bigcap \Lambda_{\vec{\xi}_j} = \emptyset, \ i \neq j \tag{3.7}$$

Figure 3.3 shows how the quadtree tessellation of the image relates to the node selection pyramid.

3.4 Region Adjacency Graph

The selected nodes R are then linked together into a graph. Although most of the following processing is done within this graph, the original quadtree structure is not abandoned: the region refinement is achieved by splitting the leaf nodes (selected nodes) into 4, maintaining the 1:4 quadtree relationships (see section 3.5.2)

3.4.1 A Neighbourhood System

A first order neighbourhood $\mathcal{N}_{\vec{\xi}}$ is defined for each selected node: two nodes are neighbours if their respective image blocks share a block edge. If the nodes in the neighbourhood system are all drawn from the same level, then this is simply the 4-neighbour system commonly used with regular pixel grids [41]. The block edge-sharing definition can be defined by considering

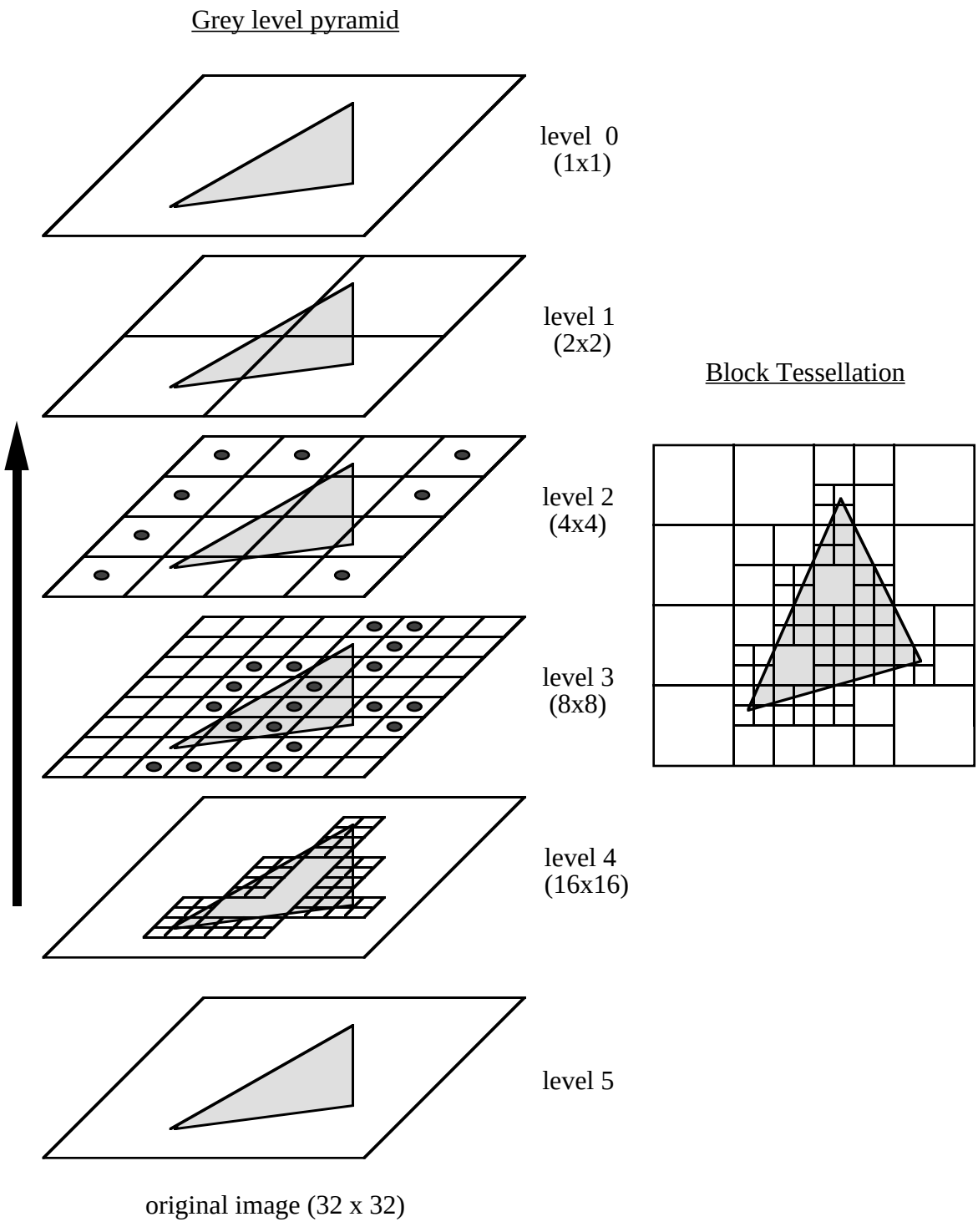


Figure 3.3: Quad-tree tessellation

the sets of pixels defining the 4 block sides (*east, north, west, south*). By making the *east* and *south* sides of the block lie 1 pixel to the right and bottom of the block it is possible to define the adjacency of two blocks

$$\begin{aligned}
\Lambda_{\vec{\xi}}^{east} &= \bigcup_{0 \le m \le 2^{M-l}} (2^{M-l} i + m, 2^{M-l}(j+1), M) \\
\Lambda_{\vec{\xi}}^{north} &= \bigcup_{0 \le n \le 2^{M-l}} (2^{M-l} i, 2^{M-l} j + n, M) \\
\Lambda_{\vec{\xi}}^{west} &= \bigcup_{0 \le m \le 2^{M-l}} (2^{M-l} i + m, 2^{M-l} j, M) \\
\Lambda_{\vec{\xi}}^{south} &= \bigcup_{0 \le n \le 2^{M-l}} (2^{M-l}(i+1), 2^{M-l} j + n, M)
\end{aligned} \tag{3.8}$$

Given the sides the nearest neighbour rule is

$$\begin{gathered}
\vec{\xi} \text{ is a neighbour of } \vec{\zeta} \text{ if} \\
(\Lambda_{\vec{\xi}}^{east} \bigcap \Lambda_{\vec{\zeta}}^{west}) \bigcup (\Lambda_{\vec{\xi}}^{west} \bigcap \Lambda_{\vec{\zeta}}^{east}) \bigcup (\Lambda_{\vec{\xi}}^{north} \bigcap \Lambda_{\vec{\zeta}}^{south}) \bigcup (\Lambda_{\vec{\xi}}^{south} \bigcap \Lambda_{\vec{\zeta}}^{north}) \neq \emptyset
\end{gathered} \tag{3.9}$$

Figure 2.6 (Chapter 2) shows a typical neighbourhood for a block in the region tessellation. The degree of each node will be at least 4 for all blocks which do not lie at the edges of the original image (the degree is at least 2 for these). Nodes from higher levels of the pyramid, representing the larger blocks, will have a greater degree than the nodes from lower levels of the pyramid.

If a set of binary valued links are created, according to the adjacency rule of equation (3.9), then each node $\vec{\xi}$ would have card$\{\mathcal{N}_{\vec{\xi}}\}$ links (the degree of the node). The set of links in this neighbourhood can be defined as all legitimate pairs

$$\mathcal{L}_{\vec{\xi}} = \bigcup_{\vec{\zeta} \in \mathcal{N}_{\vec{\xi}}} l\,(\vec{\xi}, \vec{\zeta}) \tag{3.10}$$

The complete set of links is then the set $L = \mathcal{L}_{\vec{\xi}_0} \cup \mathcal{L}_{\vec{\xi}_1} \cup \ldots \cup \mathcal{L}_{\vec{\xi}_{n-1}}$. The set of region nodes R and the set of links L then creates a modified form of a RAG $G = \{R, L\}$.

Hunter and Steiglitz [55] have described how to determine efficiently G given an inhomogeneous quadtree structure with the leaves representing the selected nodes. They term

this the *collapsing* algorithm and give a linear time complexity for finding the neighbours of a given node, and hence an overall linear time, proportional to the number of leaves, for construction of the graph. This method was implemented to generate a RAG from the selected nodes.

3.5 Iterated Decision Directed Estimation

This part of the processing is based on the Gaussian MRF signal model described in section 2.4.1, in which regions are assumed to have independently selected means. The grey level at each selected node $x(\vec{\xi})$ is iteratively averaged with its neighbours in the neighbourhood $\mathcal{N}_{\vec{\xi}}$. The value at iteration t is given by

$$x^t(\vec{\xi}) = \frac{w(\vec{\xi})x^{t-1}(\vec{\xi}) + \sum_{\vec{\zeta}\in\mathcal{N}_{\vec{\xi}}} w(\vec{\zeta})l^t(\vec{\xi},\vec{\zeta})x^{t-1}(\vec{\zeta})}{w(\vec{\xi}) + \sum_{\vec{\zeta}\in\mathcal{N}_{\vec{\xi}}} w(\vec{\zeta})l^t(\vec{\xi},\vec{\zeta})} \tag{3.11}$$

The state of the link $l(\vec{\xi},\vec{\zeta})$ acts as a switching function and the $w(\vec{\xi})$ is a weight function. The weighting is equal to the area of the image region represented by the node corresponding to the inverse of the noise variance at that node,

$$w(\vec{\xi}) = \text{card}\{\Lambda_{\vec{\xi}}\} \tag{3.12}$$

The binary link states are determined by a MAP decision rule (as outlined in section 2.5) that a given link is 'on' ($P(1)$), given the data at the two nodes it connects

$$l(\vec{\xi},\vec{\zeta}) = \begin{cases} 1 & \text{if } P_{\vec{\xi},\vec{\zeta}}(1)p(\hat{x}(\vec{\xi}) - \hat{x}(\vec{\zeta}) \mid 1) > P_{\vec{\xi},\vec{\zeta}}(0)p(\hat{x}(\vec{\xi}) - \hat{x}(\vec{\zeta}) \mid 0) \\ 0 & \text{otherwise} \end{cases} \tag{3.13}$$

where $p(\hat{x}(\vec{\xi}) - \hat{x}(\vec{\zeta}) \mid 1)$ is the conditional probability density function of the difference assuming that nodes $\vec{\xi}$ and $\vec{\zeta}$ are connected ($l(\vec{\xi},\vec{\zeta}) = 1$) i.e that the nodes belong to the same region and therefore have the same mean.

$P_{\vec{\xi},\vec{\zeta}}(1)$ may be regarded as a 'prior' of the decision process. These probabilities are calculated from estimates of edge information within the neighbourhood, thus introducing dependence on neighbouring data (see next section).

The conditional probability density of equation (3.13) can be approximated, under the assumption of the normal model for the data, by

$$p(\hat{x}(\vec{\xi}) - \hat{x}(\vec{\zeta}) \mid 1) \propto \exp\left(\frac{-(x^t(\vec{\xi}) - x^t(\vec{\zeta}))^2}{2(\overline{v}^2(\vec{\xi}) + \overline{v}^2(\vec{\zeta}))}\right) \tag{3.14}$$

$$p(\hat{x}(\vec{\xi}) - \hat{x}(\vec{\zeta}) \mid 0) \propto \exp\left(\frac{-((x^0(\vec{\xi}) - x^0(\vec{\zeta})) - (x^t(\vec{\xi}) - x^t(\vec{\zeta})))^2}{2(\overline{v}^2(\vec{\xi}) + \overline{v}^2(\vec{\zeta}))}\right) \tag{3.15}$$

It is assumed, therefore, that if the nodes $\vec{\xi}$ and $\vec{\zeta}$ belong to the same region, then their difference is noise, but if they belong to separate regions these have independently chosen means, whose difference is approximated by $(x(\vec{\xi}) - x(\vec{\zeta}))$.

3.5.1 Link Probabilities

The link probabilities of the posterior decision process of equation (3.13) are derived from modelling the neighbourhood as an edge. Given an estimate of the orientation θ, position r and probability of the edge $P(\text{edge})$ in the neighbourhood of the link $l\,(\vec{\xi},\vec{\zeta})$, the probability that a given link is 'off' $P_{\vec{\xi},\vec{\zeta}}(0)$ is calculated as a function of these parameters

$$P_{\vec{\xi},\vec{\zeta}}(0) = F(\theta, r, P(\text{edge})) \tag{3.16}$$

It has already been noted in Chapter 1 that the incorporation of edge information into a segmentation, via a 'line' process in a MRF setting, leads to problems of model identification. Also the size of the line neighbourhood is tied to the region model, which from the discussion about uncertainty, does not allow edge information from a large enough locality to influence the region process. The motivation behind the above approach is thus clear: that the parameters required to determine $P(0)$, namely the orientation, position and certainty $[\theta, r, P(\text{edge})]$, are directly available from any boundary information, and if

any independent, region-based estimate of orientation can be made, then it would allow the information between the region and boundary processes to be combined.

The probability of the link state given the data in the neighbourhood may be written as the probability of the link state given the parameters of an edge in that neighbourhood, if it is assumed that the estimated edge is close to the real edge in a MMSE sense:

$$P(l\,(\vec{\xi},\vec{\zeta}) \mid x(\vec{\eta}), \vec{\eta} \in \mathcal{N}_{\vec{\xi}}) = P(l\,(\vec{\xi},\vec{\zeta}) \mid f_{\vec{\eta}}(\theta_{\vec{\xi}}, r_{\vec{\xi}}), \epsilon_{\vec{\eta}}(\theta_{\vec{\xi}}, r_{\vec{\xi}}), \vec{\eta} \in \mathcal{N}_{\vec{\xi}}) \tag{3.17}$$

where $l\,(\vec{\xi},\vec{\zeta})$ is the link state, $f_{\vec{\eta}}(\theta_{\vec{\xi}}, r_{\vec{\xi}})$, is the model of the data in the neighbourhood, given a MMSE estimate of the edge parameters and $\epsilon_{\vec{\eta}}(\theta_{\vec{\xi}}, r_{\vec{\xi}})$ are the residuals: $x(\vec{\eta}) - f_{\vec{\eta}}(\theta_{\vec{\xi}}, r_{\vec{\xi}}), \vec{\eta} \in \mathcal{N}_{\vec{\xi}}$. Manipulating the r.h.s of equation (3.17)

$$\begin{aligned}
& P(l\,(\vec{\xi},\vec{\zeta}) \mid f_{\vec{\eta}}(\theta_{\vec{\xi}}, r_{\vec{\xi}}), \epsilon_{\vec{\eta}}(\theta_{\vec{\xi}}, r_{\vec{\xi}}), \vec{\eta} \in \mathcal{N}_{\vec{\xi}}) \\
= \;& \frac{p(l\,(\vec{\xi},\vec{\zeta}), \epsilon_{\vec{\eta}}(\theta_{\vec{\xi}}, r_{\vec{\xi}}) \mid f_{\vec{\eta}}(\theta_{\vec{\xi}}, r_{\vec{\xi}}))}{p(\epsilon_{\vec{\eta}}(\theta_{\vec{\xi}}, r_{\vec{\xi}}) \mid f_{\vec{\eta}}(\theta_{\vec{\xi}}, r_{\vec{\xi}}))} \\
= \;& \frac{P(l\,(\vec{\xi},\vec{\zeta}) \mid f_{\vec{\eta}}(\theta_{\vec{\xi}}, r_{\vec{\xi}})) \times p(\epsilon_{\vec{\eta}}(\theta_{\vec{\xi}}, r_{\vec{\xi}}) \mid f_{\vec{\eta}}(\theta_{\vec{\xi}}, r_{\vec{\xi}}))}{p(\epsilon_{\vec{\eta}}(\theta_{\vec{\xi}}, r_{\vec{\xi}}) \mid f_{\vec{\eta}}(\theta_{\vec{\xi}}, r_{\vec{\xi}})} \\
= \;& P(l\,(\vec{\xi},\vec{\zeta}) \mid f_{\vec{\eta}}(\theta_{\vec{\xi}}, r_{\vec{\xi}}))
\end{aligned} \tag{3.18}$$

(where $\vec{\eta} \in \mathcal{N}_{\vec{\xi}}$). The second step follows if the residuals are noise and so independent of the link state, under the assumption that the edge is a MMSE estimate.

After a description of how the prior link probabilities $P(0)$ can be calculated from a block edge estimate $[\theta, r, P(\text{edge})]$, an orientation estimation method is described that operates within the region block tessellation.

By making the simplifying assumption that the p.d.f. of the position is Gaussian, the probability that an edge intersects the link can be derived. Given two connected region nodes $\vec{\xi}$ and $\vec{\zeta}$ and the link between them of length l and angle ϕ i.e. $l\,(\vec{\xi},\vec{\zeta}) = [l, \phi]$, and the estimate of the edge in the block $[\theta, r, P(\text{edge})]$, the probability that the link is 'off', is

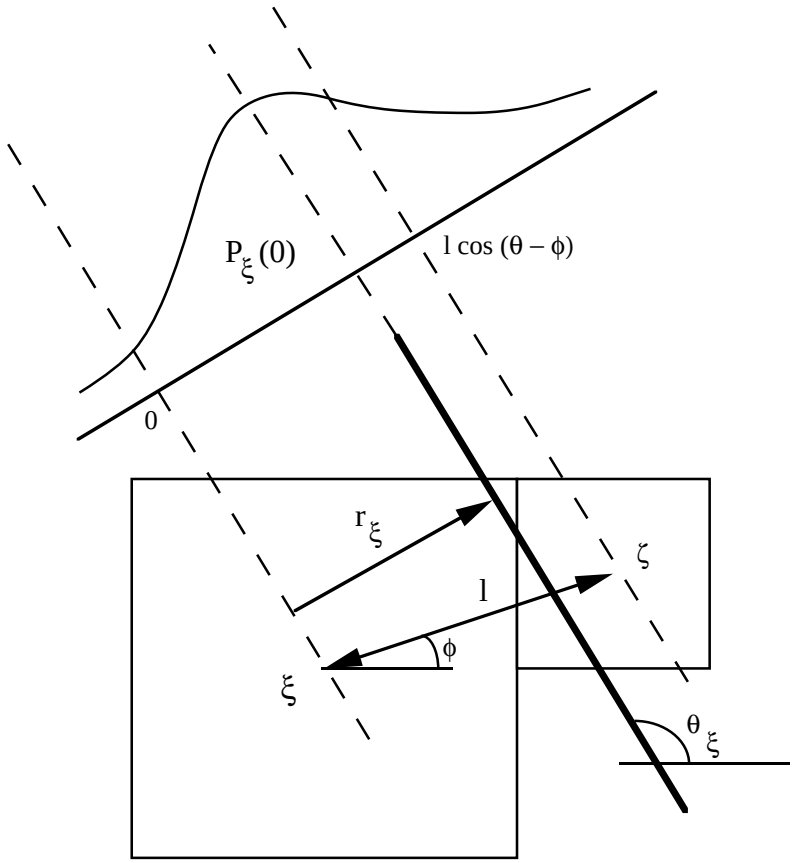


Figure 3.4: $P(0)$ given the block orientation estimate

given by

$$P_{\vec{\xi},\vec{\zeta}}(0) = P_{\vec{\xi}}(\text{edge})\frac{1}{\sqrt{2\pi}\sigma_{r_{\vec{\xi}}}}\int_0^{l\cos(\phi-\theta_{\vec{\xi}})}\exp\left(-\frac{(r-r_{\vec{\xi}})^2}{2\sigma^2_{r_{\vec{\xi}}}}\right)dr \tag{3.19}$$

where the p.d.f. of the edge within the block has the positional variance $\sigma^2_{r_{\vec{\xi}}}$ and is modelled as a Gaussian. The derivation of the limits of the integral is illustrated for a typical example in figure 3.4.

Equation 3.19 can be simplified by substitution

$$\begin{aligned} P_{\vec{\xi},\vec{\zeta}}(0) &= P_{\vec{\xi}}(\text{edge})\frac{1}{\sqrt{2\pi}}\int_{\frac{r_{\vec{\xi}}}{\sigma_{r_{\vec{\xi}}}}}^{\frac{r_{\vec{\xi}}}{\sigma_{r_{\vec{\xi}}}}-\frac{l\cos(\phi-\theta_{\vec{\xi}})}{\sigma_{r_{\vec{\xi}}}}}\exp\left(-\frac{u^2}{2}\right)du \\ &= P_{\vec{\xi}}(\text{edge})\left[\text{erfc}\left(\frac{l\cos(\phi-\theta_{\vec{\xi}})-r_{\vec{\xi}}}{\sigma_{r_{\vec{\xi}}}}\right)-\text{erfc}\left(-\frac{r_{\vec{\xi}}}{\sigma_{r_{\vec{\xi}}}}\right)\right] \end{aligned} \tag{3.20}$$

where erfc() is the complementary error function [75] defined by

$$\text{erfc}(x) = 1 - \text{erf}(x) \tag{3.21}$$

where

$$\text{erf}(x) = \frac{1}{\sqrt{2\pi}} \int_0^x \exp\left(-\frac{y^2}{2}\right) dy \tag{3.22}$$

The positional variance for the block orientation estimate $\sigma^2_{r_{\vec{\xi}}}$ can be calculated at the same time as the estimate itself (see below). In general, however, this will remain constant over scales as for larger blocks there is poorer spatial resolution and with smaller blocks there is greater noise affecting the estimate.

As each region link connects two nodes, the link probability may be different when calculated from either node. This symmetry problem is solved by taking the maximum of the two probabilities, which is a logical OR operation, thus allowing the edge to be seen from either block.

In summary then, the region estimation link decision process is reliant on having a good estimate of the orientation, position and certainty of any boundary in the block. This overcomes the need of a separate 'line' process commonly adopted by researchers employing MRF methods (e.g. [39] [38]), and the problems this presents as far as specifying the prior probabilities for conditional link configurations.

Block Based Region Orientation Estimation

The basis of the above region link decision process is the block based orientation estimate $[\theta, r, P(\text{edge})]$. The estimation method described in the following section is iterative and operates only within the block neighbourhood $\mathcal{N}_{\vec{\xi}}$ already defined. First, an estimate of the orientation in the block is obtained then, using a simple piecewise linear model of an edge in the block, a weighted least squares method is used to determine the position and amplitude

of the edge. Finally, the estimates are combined by averaging in the neighbourhoods to reduce orientation bias, giving the estimates a full 8-neighbourhood.

Edge Orientation, Amplitude and Position Estimation

An initial orientation estimate for the block is calculated by

$$\theta_{\vec{\xi}} = \arg\left(\sum_{\vec{\zeta}\in\mathcal{N}_{\vec{\xi}}} \left(x(\vec{\xi}) - x(\vec{\zeta})\right)^2 \mathbf{l}\left(\vec{\xi},\vec{\zeta}\right)\right) + \frac{\pi}{2} \tag{3.23}$$

where $\mathbf{l}\left(\vec{\xi},\vec{\zeta}\right)$ is the unit vector in the direction of the link and the sum is weighted by the squared differences in grey level. The $\frac{\pi}{2}$ term is necessary as the resultant vector of the sum is normal to the required estimate. There are two sources of bias in this estimate: the first arises from the assumption that the edge runs through the block centre (implicit in the links being defined as joining block centres) and the second from not taking into account the corner blocks which would result in a full 8-neighbourhood system. The former bias is reduced by estimating a centroid for the block using centres of the neighbouring blocks, again weighted by the squared errors, and then using this as the origin for the above orientation estimate. The latter bias is reduced by the neighbourhood averaging step.

To obtain an estimate of the amplitude and position the edge is modelled as a piecewise linear function of position. The grey level of the neighbours is modelled by

$$\hat{x}_{\vec{\zeta}} = a_{\vec{\xi}} f_{\vec{\zeta}}(r_{\vec{\xi}}) + b_{\vec{\xi}} \tag{3.24}$$

where $[a_{\vec{\xi}}, b_{\vec{\xi}}]$ are the amplitude and mean of the edge running through the neighbourhood of $\vec{\xi}$, at an offset $r_{\vec{\xi}}$ (normal to the edge orientation) from the block centre, (figure 3.5). The ramp edge is a piecewise linear function

$$f_{\vec{\zeta}}(r_{\vec{\xi}}) = f\left[\left(\left(i_{\vec{\xi}} - i_{\vec{\zeta}}\right)\cos\theta_{\vec{\xi}}^{\perp} + \left(j_{\vec{\xi}} - j_{\vec{\zeta}}\right)\sin\theta_{\vec{\xi}}^{\perp} + r_{\vec{\xi}}\right)/\sqrt{2}l_{\vec{\zeta}}\right] \tag{3.25}$$

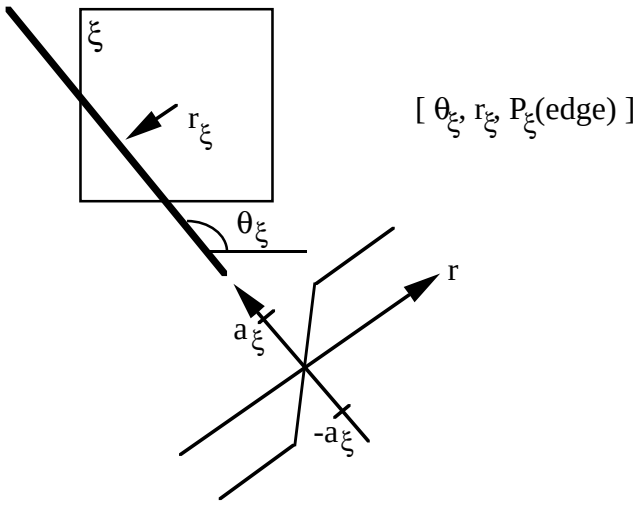


Figure 3.5: Region block edge model

where (i, j) are the region block centres and l is the region block size. (Note that the normal to the estimated orientation is used denoted by the superscript $\perp$), where f is the piecewise linear function

$$f(x) = \begin{cases} -1 & x < 1 \\ x & |x| \leq 1 \\ 1 & x > 1 \end{cases} \tag{3.26}$$

Solving for a, b can be treated as a weighted least squares minimisation problem

$$\epsilon = \sum_{\vec{\zeta} \in \mathcal{N}_{\vec{\xi}}} \left(x(\vec{\zeta}) - \hat{x}(\vec{\zeta})\right)^2 l_{\vec{\zeta}}^2 \tag{3.27}$$

where the weighting is the area of the neighbourhood block and reflects the certainty of the data. Assuming that an initial estimate of the orientation $\theta_{\vec{\xi}}^{\perp}$ and position $r_{\vec{\xi}}$ are given, then:

1. $a_{\vec{\xi}}, b_{\vec{\xi}}$ are given by

$$\begin{aligned} \frac{\partial \epsilon}{\partial a_{\vec{\xi}}} &= 0 \\ \Rightarrow a_{\vec{\xi}} &= \frac{\sum_{\vec{\zeta}} f_{\vec{\zeta}} x(\vec{\zeta}) l_{\vec{\zeta}}^2 - b_{\vec{\xi}} \sum_{\vec{\zeta}} f_{\vec{\zeta}} l_{\vec{\zeta}}^2}{\sum_{\vec{\zeta}} f_{\vec{\zeta}}^2 l_{\vec{\zeta}}^2} \end{aligned} \tag{3.28}$$

(subscripts have been deliberately left-out for clarity). Note that the neighbourhood

is augmented with the data at the block under consideration (i.e. $\vec{\xi}$).

$$\begin{aligned} \frac{\partial \epsilon}{\partial b_{\vec{\xi}}} &= 0 \\ \Rightarrow b_{\vec{\xi}} &= \frac{\sum_{\vec{\zeta}} l_{\vec{\zeta}}{}^2 (x(\vec{\zeta}) - a_{\vec{\xi}} f_{\vec{\zeta}})}{\sum_{\vec{\zeta}} l_{\vec{\zeta}}{}^2} \end{aligned} \tag{3.29}$$

Substituting for $b_{\vec{\xi}}$ in equation (3.28) yields

$$a_{\vec{\xi}} = \frac{\sum_{\vec{\zeta}} f_{\vec{\zeta}} x(\vec{\zeta}) l_{\vec{\zeta}}{}^2 \sum_{\vec{\zeta}} l_{\vec{\zeta}}{}^2 - \sum_{\vec{\zeta}} l_{\vec{\zeta}}{}^2 x(\vec{\zeta}) \sum_{\vec{\zeta}} f_{\vec{\zeta}} l_{\vec{\zeta}}{}^2}{\sum_{\vec{\zeta}} l_{\vec{\zeta}}{}^2 \sum_{\vec{\zeta}} f_{\vec{\zeta}}{}^2 l_{\vec{\zeta}}{}^2 - \left(\sum_{\vec{\zeta}} f_{\vec{\zeta}} l_{\vec{\zeta}}{}^2\right)^2} \tag{3.30}$$

2. Knowing $a_{\vec{\xi}}$ and $b_{\vec{\xi}}$ and an expression for $r_{\vec{\xi}}$ is given by the following

$$\begin{aligned} \frac{\partial \epsilon}{\partial r_{\vec{\xi}}} &= 0 \\ \Rightarrow \sum_{\vec{\zeta}} \frac{\partial f_{\vec{\zeta}}}{\partial r_{\vec{\xi}}} x(\vec{\zeta}) l_{\vec{\zeta}}{}^2 &= a_{\vec{\xi}} \sum_{\vec{\zeta}} \frac{\partial f_{\vec{\zeta}}}{\partial r_{\vec{\xi}}} f_{\vec{\zeta}} l_{\vec{\zeta}}{}^2 + b_{\vec{\xi}} \sum_{\vec{\zeta}} \frac{\partial f_{\vec{\zeta}}}{\partial r_{\vec{\xi}}} l_{\vec{\zeta}}{}^2 \end{aligned} \tag{3.31}$$

Now from equations (3.25) and (3.26)

$$\begin{aligned} \frac{\partial f_{\vec{\zeta}}}{\partial r_{\vec{\xi}}} &= 0, \text{ if } \left| (i_{\vec{\xi}} - i_{\vec{\zeta}}) \cos\theta^{\perp}_{\vec{\xi}} + (j_{\vec{\xi}} - j_{\vec{\zeta}}) \sin\theta^{\perp}_{\vec{\xi}} + r_{\vec{\xi}} \right| > \sqrt{2} l_{\vec{\zeta}} \\ &= -\frac{1}{\sqrt{2} l_{\vec{\zeta}}}, \text{ else} \end{aligned} \tag{3.32}$$

then by considering the sum over the neighbours that lie within part of the modelled edge where the gradient is non-zero ($\mathcal{N}'_{\vec{\xi}}$), equation (3.31) becomes

$$\begin{aligned} \sum_{\vec{\zeta} \in \mathcal{N}'_{\vec{\xi}}} x(\vec{\zeta}) l_{\vec{\zeta}} &= a_{\vec{\xi}} \sum_{\vec{\zeta} \in \mathcal{N}'_{\vec{\xi}}} l_{\vec{\zeta}} f_{\vec{\zeta}} + b_{\vec{\xi}} \sum_{\vec{\zeta} \in \mathcal{N}'_{\vec{\xi}}} l_{\vec{\zeta}} \\ \Rightarrow \sum_{\vec{\zeta} \in \mathcal{N}'_{\vec{\xi}}} x(\vec{\zeta}) l_{\vec{\zeta}} &= a_{\vec{\xi}} \sum_{\vec{\zeta} \in \mathcal{N}'_{\vec{\xi}}} \left[(i_{\vec{\xi}} - i_{\vec{\zeta}}) \cos\theta^{\perp}_{\vec{\xi}} + (j_{\vec{\xi}} - j_{\vec{\zeta}}) \sin\theta^{\perp}_{\vec{\xi}} + r_{\vec{\xi}} \right] / \sqrt{2} + b_{\vec{\xi}} \sum_{\vec{\zeta} \in \mathcal{N}'_{\vec{\xi}}} l_{\vec{\zeta}} \\ \Rightarrow r_{\vec{\xi}} &= \frac{1}{\mathrm{card}\mathcal{N}'_{\vec{\xi}}} \frac{\sqrt{2}}{a_{\vec{\xi}}} \sum_{\vec{\zeta}} (x(\vec{\zeta}) - b_{\vec{\xi}}) l_{\vec{\zeta}} \\ &\quad + \sum_{\vec{\zeta}} \left((i_{\vec{\xi}} - i_{\vec{\zeta}}) \cos\theta^{\perp}_{\vec{\xi}} + (j_{\vec{\xi}} - j_{\vec{\zeta}}) \sin\theta^{\perp}_{\vec{\xi}} \right) \end{aligned} \tag{3.33}$$

```
procedure Edge_Estimation (Theta, r)

begin
        i = 0;
        a[0] = 0.0;
        b[0] = 0.0;
        r[0] = r;
        repeat
                i = i + 1;
                calculate a[i] and b[i]; {step 1}
                r[i] from a[i], b[i];  {step 2}
        until abs(r[i] - r[i-1]) < 0.5
        return (a[i], b[i], r[i])
end
```

Figure 3.6: Iterative edge parameter estimation

3. The variance of the amplitude is estimated by considering it as a linear combination of the block variances of the neighbourhood

$$\begin{aligned} \mathrm{var}a_{\vec{\xi}} &= \sum_{\vec{\zeta}} \alpha^2_{\vec{\xi},\vec{\zeta}} \overline{v}^2_{\vec{\zeta}} \\ \Rightarrow \alpha_{\vec{\xi},\vec{\zeta}} &= \frac{\left[f_{\vec{\zeta}} \sum_{\vec{\eta}} l^2_{\vec{\eta}} - \sum_{\vec{\eta}} f_{\vec{\eta}}\right] l^2_{\vec{\eta}}}{\sum_{\vec{\eta}} l^2_{\vec{\eta}} \sum_{\vec{\eta}} {f_{\vec{\eta}}}^2 {l_{\vec{\eta}}}^2 - \left(\sum_{\vec{\eta}} f_{\vec{\eta}} {l_{\vec{\eta}}}^2\right)^2} \end{aligned} \tag{3.34}$$

4. The variance of the position is derived from squaring the first term of equation (3.33) and replacing the edge model estimate minus the mean by the block variance estimates

$$\mathrm{var}r_{\vec{\xi}} \approx \frac{2 \sum_{\vec{\zeta}} \overline{v}^2_{\vec{\zeta}} {l_{\vec{\eta}}}^2}{\mathrm{card}\{\mathcal{N}'_{\vec{\xi}}\}^2 a^2_{\vec{\xi}}} \tag{3.35}$$

The estimation procedure is summarised by the pseudo-code given in figure 3.6 where the input parameters are the initial estimates $[\theta_{\vec{\xi}}, r_{\vec{\xi}}]$. Note that the process is repeated until the position estimate has converged to the required accuracy (< 0.5 pixel). In practice, this iterative estimation process converges rapidly, within 3 to 5 iterations, giving good estimates of $r_{\vec{\xi}}$. The edge amplitude and position estimation is performed once for all new

region nodes (the initial tessellation nodes are all classed as being 'new'), including the nodes created by the block splitting process described in section 3.5.2.

Neighbourhood Averaging

The estimates of $[\theta_{\vec{\xi}}, a_{\vec{\xi}}, r_{\vec{\xi}}]$ are combined with those of the neighbours of $\vec{\xi}$. New $[\theta_{\vec{\xi}}, a_{\vec{\xi}}]$ are obtained by averaging such that

$$a'_{\vec{\xi}} \begin{pmatrix} \cos 2\theta'_{\vec{\xi}} \\ \sin 2\theta'_{\vec{\xi}} \end{pmatrix} = \sum_{\vec{\zeta} \in \mathcal{N}_{\vec{\xi}} \cup \{\vec{\xi}\}} a_{\vec{\zeta}} \begin{pmatrix} \cos 2\theta_{\vec{\zeta}} \\ \sin 2\theta_{\vec{\zeta}} \end{pmatrix} \tag{3.36}$$

The doubling of the angle is essential to overcome the sign ambiguity of the orientation [61], so that θ and $\theta + \pi$ are regarded as the same estimate. The whole question of orientation representation is discussed in greater detail in Chapter 4. The best combined estimate of position can be computed using the amplitude values to a second weighted least squares estimation

$$\epsilon = \sum_{\vec{\zeta} \in \mathcal{N}_{\vec{\xi}} \cup \{\vec{\xi}\}} \left(r_{\vec{\zeta}} - r_{\vec{\xi}} + (i_{\vec{\zeta}} - i_{\vec{\xi}}) \cos\theta^{\perp}_{\vec{\xi}} + (j_{\vec{\zeta}} - j_{\vec{\xi}}) \sin\theta^{\perp}_{\vec{\xi}} \right)^2 a^2_{\vec{\zeta}} \tag{3.37}$$

the minimum w.r.t. $r_{\vec{\xi}}$ is then

$$\begin{aligned} \frac{\partial \epsilon}{\partial r_{\vec{\xi}}} &= 0 \\ \Rightarrow r_{\vec{\xi}} + i_{\vec{\xi}} \cos\theta^{\perp}_{\vec{\xi}} + j_{\vec{\xi}} \sin\theta^{\perp}_{\vec{\xi}} &= \frac{\sum_{\vec{\zeta} \in \mathcal{N}_{\vec{\xi}} \cup \{\vec{\xi}\}} a^2_{\vec{\zeta}} \left(r_{\vec{\zeta}} + i_{\vec{\zeta}} \cos\theta^{\perp}_{\vec{\xi}} + j_{\vec{\zeta}} \sin\theta^{\perp}_{\vec{\xi}} \right)}{\sum_{\vec{\zeta} \in \mathcal{N}_{\vec{\xi}} \cup \{\vec{\xi}\}} a^2_{\vec{\zeta}}} \end{aligned} \tag{3.38}$$

So the best combined estimate is simply the centroid of the position vectors weighted by the squares of the block amplitudes. This neighbourhood averaging is performed at every iteration of the region estimation process.

An Edge Certainty Measure

The question remains of what is the probability of an edge $P(\text{edge})$, based on the block data. Since $a_{\vec{\xi}}$ is an estimate of edge amplitude, it seems reasonable to consider this against

the estimate of the noise in the block. The form of certainty measure used is the conditional probability $P(\text{edge} \mid \text{amplitude})$, which by Bayes's theorem is

$$P(\text{edge} \mid a_{\vec{\xi}}) = \frac{p(a_{\vec{\xi}} \mid \text{edge})P(\text{edge})}{p(a_{\vec{\xi}} \mid \text{edge})P(\text{edge}) + p(a_{\vec{\xi}} \mid \overline{\text{edge}})P(\overline{\text{edge}})} \tag{3.39}$$

which leaves the specification of the p.d.f. $p(a_{\vec{\xi}} \mid \text{edge})$ and the prior probability of an edge in any given block. Assuming that edges are equally likely to be positive or negative and have average amplitude A, the density of $a_{\vec{\xi}}$ is a mixture of the form

$$\begin{aligned} p(a_{\vec{\xi}} \mid \text{edge}) &= \frac{0.5}{\sqrt{2\pi}\sigma_{\vec{\xi}}} \left[\exp\left(-\frac{(a_{\vec{\xi}} - A)^2}{2\sigma_{\vec{\xi}}^2} \right) + \exp\left(-\frac{(a_{\vec{\xi}} + A)^2}{2\sigma_{\vec{\xi}}^2} \right) \right] \\ &= \frac{1}{\sqrt{2\pi}\sigma_{\vec{\xi}}} \exp\left(-\frac{a_{\vec{\xi}}^2}{2\sigma_{\vec{\xi}}^2} - \frac{A^2}{2} \right) \cosh\left(\frac{A}{\sigma_{\vec{\xi}}} \right) \end{aligned} \tag{3.40}$$

and

$$p(a_{\vec{\xi}} \mid \overline{\text{edge}}) = \frac{1}{\sqrt{2\pi}\sigma_{\vec{\xi}}} \exp\left(-\frac{a_{\vec{\xi}}^2}{2\sigma_{\vec{\xi}}^2} \right) \tag{3.41}$$

Substituting from equation (3.40) and (3.41) into equation (3.39)

$$P(\text{edge} \mid a_{\vec{\xi}}) = \frac{P(\text{edge}) \cosh(\frac{A}{\sigma_{\vec{\xi}}})}{P(\text{edge}) \cosh(\frac{A}{\sigma_{\vec{\xi}}}) + \exp(\frac{A^2}{2})} \tag{3.42}$$

Figure 3.7 shows a normalised form of equation (3.42) with $P(\text{edge}) = 0.5$ and $A = 3\sigma_{\vec{\xi}}$, which was used in the experiments presented at the end of this chapter. The edge certainty is an important measure as it serves two purposes. As well as influencing the region link decision processes through equation (3.19), it is also used to decide whether a region block should be split: any block which has a high certainty of an edge within it becomes a candidate for splitting as it is no longer seen adequately to represent one region, as described below.

3.5.2 Region Refinement

Running the estimation algorithm for several iterations without any attempt to refine the initial block segmentation stabilises quickly, producing more or less 'closed' regions. There

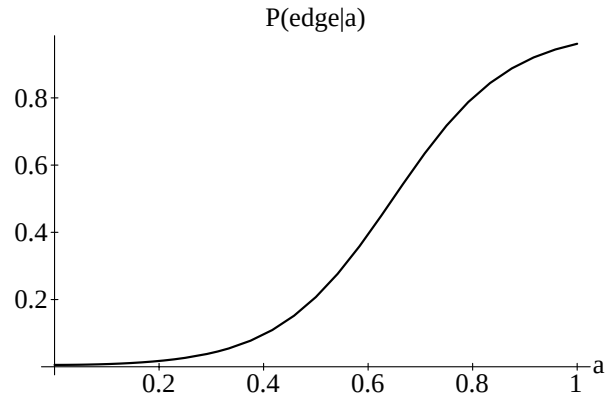


Figure 3.7: Normalised edge certainty measure

is, however, no guarantee that this process will converge to a satisfactory result. In fact, since some blocks around the boundaries will be a mixture of more than one region, they are likely to favour being linked equally to both the object and the background regions. These 'holes' in the boundary may 'leak' into the background leading to the highly undesirable convergence to a uniform grey level. Some improvement of spatial resolution, while maintaining class resolution, is therefore required.

The requirements of such a refinement process can be stated as:

1. To perform local, recursive, top-down refinement of the regions, with the aim of obtaining better position resolution.

2. It would be advantageous to keep the existing quadtree tessellation by splitting blocks into 4 i.e. *spawning* children to the leaves of the initial tree, and relinking these into the region adjacency graph G.

3. Such a process would (obviously) have to concentrate on the non-interior blocks, hence defining a boundary region or region of uncertainty. This implies a test to say which

blocks are non-interior (*edge*) and whether they should be split.

The region refinement process used shares some aspects of the boundary estimation described by Spann and Wilson [88], but has some notable differences. Having outlined the nature and requirements of the processing required, what then is to be expected at the higher spatial resolutions, below the lowest level of the initial block tessellation? The noise will be greater and therefore the data unreliable to make link decisions, without the use of the region data already estimated. In the Spann and Wilson algorithm this was achieved by smoothing of the noisy data followed by a maximum likelihood classification into the region classes present at the level above. A similar mechanism is needed to 'focus' the classes present at the parent level.

A Criterion for Block Splitting

The criterion for block splitting takes the form of a hypothesis test based on the posterior probability ratio

$$p(\text{children} \mid \text{edge})P(\text{edge}) \underset{>}{\overset{?}{}} p(\text{children} \mid \overline{\text{edge}})P(\overline{\text{edge}}) \tag{3.43}$$

The hypotheses are:

H_0**:** The block $\vec{\xi}$ is interior so that its children $\{\vec{\xi}_0, \vec{\xi}_1, \vec{\xi}_2, \vec{\xi}_3\}$ are interior and therefore

$$p(\vec{\xi}_k \mid \overline{\text{edge}}) = \exp\left(\sum_k -\frac{\left(\hat{x}^I(\vec{\xi}_k) - x(\vec{\xi}_k)\right)^2}{2\overline{v}_{\vec{\xi}^2}}\right) \tag{3.44}$$

H_1**:** The block has an edge running through it

$$p(\vec{\xi}_k \mid \text{edge}) = \exp\left(\sum_k -\frac{\left(\hat{x}^E(\vec{\xi}_k) - x(\vec{\xi}_k)\right)^2}{2\overline{v}_{\vec{\xi}^2}}\right) \tag{3.45}$$

For the null hypothesis, the estimates for the children are set to be the same as the parent i.e $\hat{x}^I(\vec{\xi}_k) = x(\vec{\xi}), 0 \leq k < 4$. For the edge hypothesis the estimates for the children are chosen

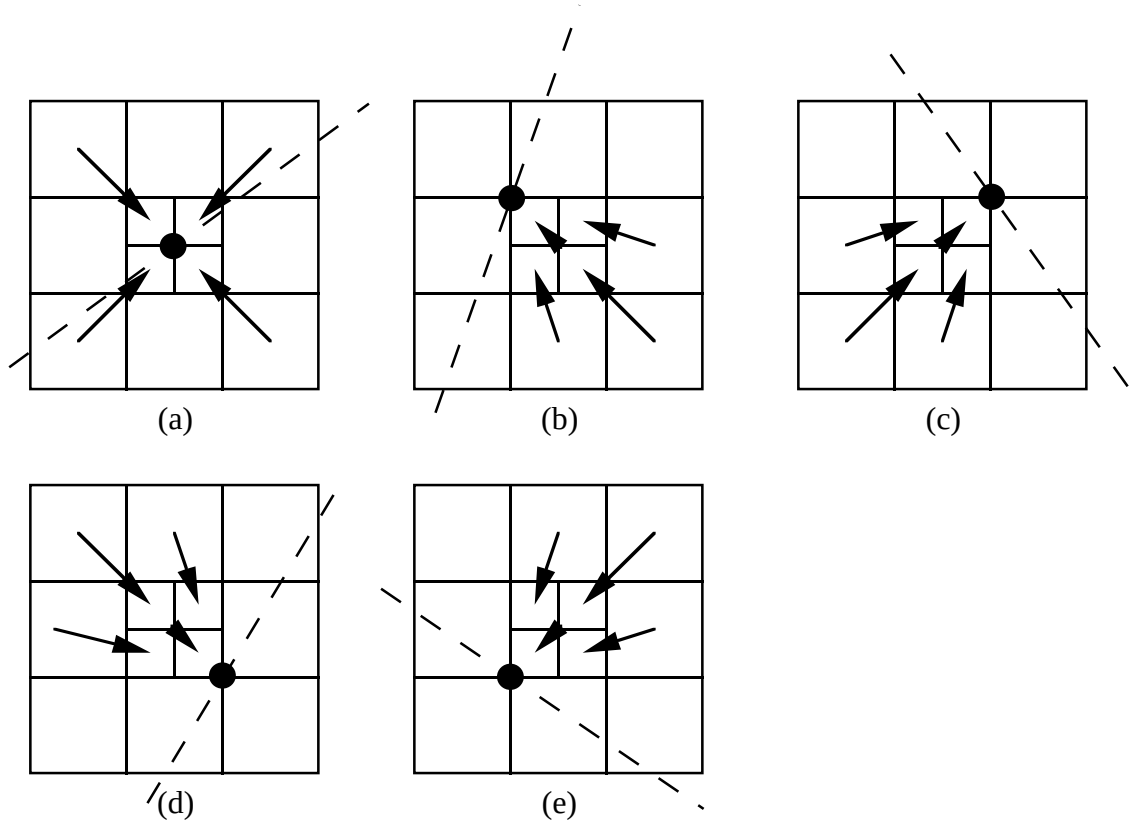


Figure 3.8: Spatial shrinking transformations

from the data values in the *neighbourhood* $\mathcal{N}_{\vec{\xi}}$. The chosen set or configuration depends on the current estimate of $[\theta_{\vec{\xi}}, r_{\vec{\xi}}]$.

'Focusing' Edge Regions

For a straight edge, the grey level values can be represented by some combination of the grey level values of the neighbourhood of the parent. Figure 3.8 illustrates this for a regular 3×3 neighbourhood where the parent node at the centre has been replaced by its children taking the values from $\mathcal{N}_{\vec{\xi}}$. For configuration (a), where the edge passes through the block centre, it can easily be verified that taking the 4 *corner* neighbours of $\vec{\xi}$ provides a suitable approximation of the actual values of the children, independent of the orientation of the edge. Configurations (b) through to (e) are required for the 4 cases where the edge passes

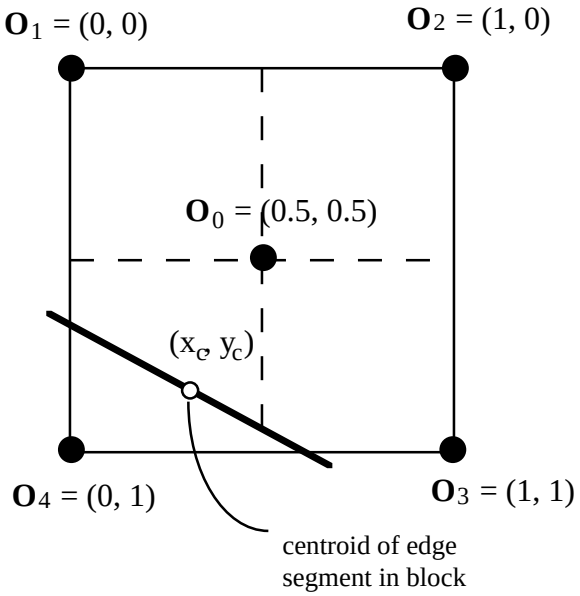


Figure 3.9: Spatial dilation origins

to one side of the block centre. These configurations represent a set of spatial dilations with the origin either at the block centre or at the 4 corners. They are similar to the fractal *contraction* transformations used in block based fractal coding [57] [6] [97]. In this application, they neatly effect the focusing of the region edge information which is the aim of the block splitting.

Selection of one of the configurations shown in figure 3.8 is determined by the position of the transformation origin in relation to the centroid of the block edge estimate $[\theta_{\vec{\xi}}, r_{\vec{\xi}}]$. The probability of selection is approximated by

$$W(i) = (1.0 - |o_{i0} - x_c|)(1.0 - |o_{i1} - y_c|) \tag{3.46}$$

where (x_c, y_c) is the centroid of the edge segment in the block and $\vec{o}_i$ is the position of the origin of configuration i, as shown in figure 3.9. The test of equation (3.43) is only performed for blocks for which $P(\text{edge}) > 0.5$ which reduces the computational burden to the set of boundary blocks only. Having decided to split a block into 4, the values of the children $\hat{x}^E(\vec{\xi}_k)$ are taken from the appropriate configuration and the child data variance set

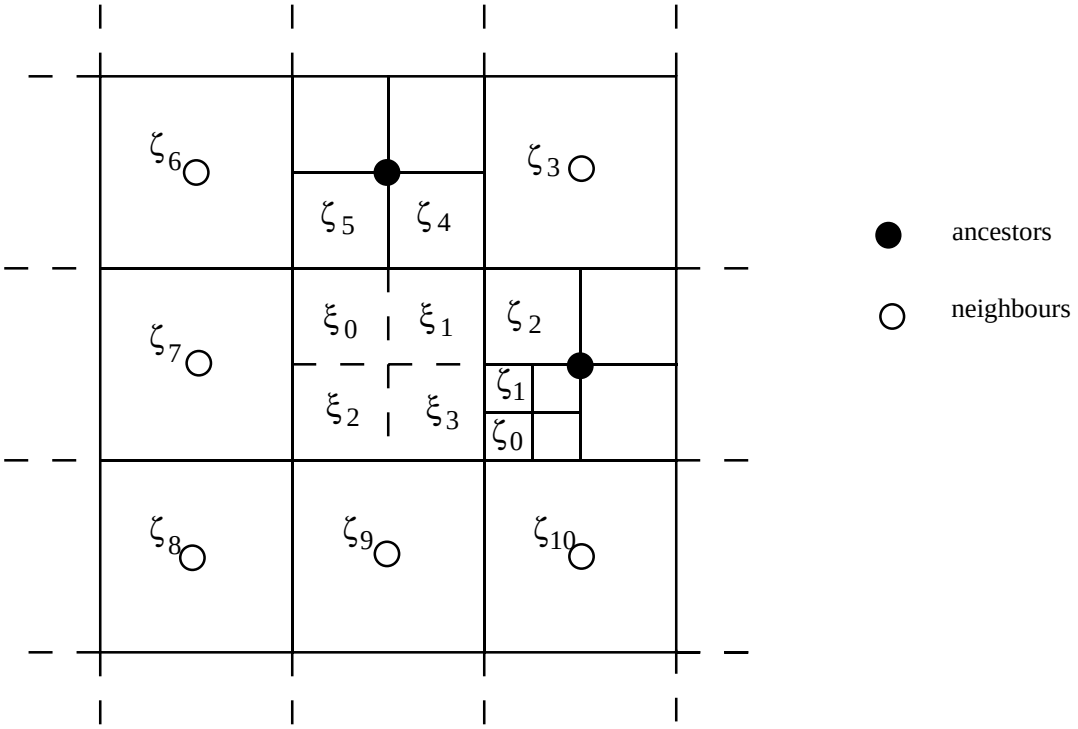


Figure 3.10: Block splitting in a non-regular neighbourhood

to be equal to $\overline{v}^2_{\vec{\xi}}$. The newly spawned children are then linked into the current tessellation using the same neighbourhood adjacency criteria as in the initial block tessellation (see section 3.4.1).

The illustrations in figure 3.8 are artificial, as in general the tessellation will not be regular. To ensure that only reliable data is used for the child estimates, the neighbour data $x(\vec{\zeta})$ is only used if it is a sibling or an ancestor of $x(\vec{\xi})$. If $x(\vec{\zeta})$ happens to be at a lower level then the quadtree is ascended until the same level as $x(\vec{\xi})$ is reached. Figure 3.10 depicts a more typical situation where the neighbours $\{\vec{\zeta_0}, \vec{\zeta_1}, \vec{\zeta_2}, \vec{\zeta_4}, \vec{\zeta_5}\}$ are at lower levels so their ancestors are used (marked by small circles), to form the appropriate configuration for the H_1 test.

Vertical Propagation of Edge Estimation Information

The children that are spawned after each iteration are regarded as being new and the initial orientation estimation is performed on them. The averaging of the estimates takes place as before in the new neighbourhood. The orientation estimates of the now redundant parent block, however, are not discarded. The new estimates of the children are combined with the parent, thus propagating down the more reliable orientation information from the higher levels of the quadtree. The combined estimates $\theta'_{\vec{\xi}_k}$ and $\mathbf{r}'_{\vec{\xi}_k}$ are given by

$$\begin{aligned} \theta'_{\vec{\xi}_k} &= \beta_c \theta_{\vec{\xi}} + \beta_p \theta_{\vec{\xi}} \\ \mathbf{r}'_{\vec{\xi}_k} &= \beta_c \mathbf{r}_{\vec{\xi}_k} + \beta_p \mathbf{r}_{\vec{\xi}} \end{aligned} \tag{3.47}$$

where the coefficients are derived from the respective variances of the amplitudes of the block estimates (equation (3.34))

$$\begin{aligned} \beta_c &= \frac{\mathrm{var} a_{\vec{\xi}_k}}{\mathrm{var} a_{\vec{\xi}_k} + \mathrm{var} a_{\vec{\xi}}} \\ \beta_p &= \frac{\mathrm{var} a_{\vec{\xi}}}{\mathrm{var} a_{\vec{\xi}_k} + \mathrm{var} a_{\vec{\xi}}} \end{aligned} \tag{3.48}$$

The variance estimate itself is updated using the same coefficients such that

$$\mathrm{var} a'_{\vec{\xi}_k} = \beta_c \mathrm{var} a_{\vec{\xi}_k} + \beta_p \mathrm{var} a_{\vec{\xi}} \tag{3.49}$$

which ensures that further splitting will continue to propagate the estimates towards the image plane. The vertical propagation is adaptive to the noise, so that if the child estimates are as good as or better than that of the parent then more weighting will be given to them, so it is not always the case that the estimates from higher levels will dominate the process.

3.5.3 Convergence Criteria

There are two stopping conditions that were investigated:

1. Stop when there are no changes in link states between consecutive iterations.

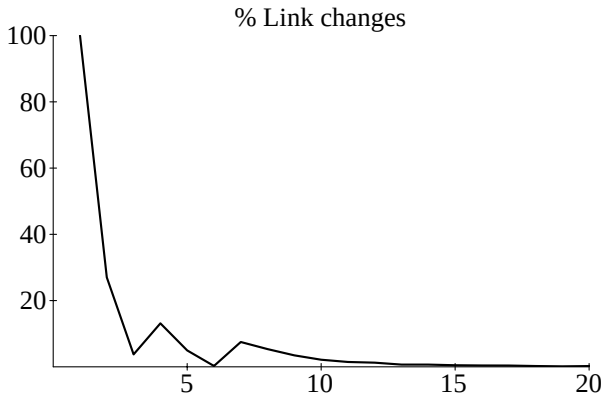


Figure 3.11: Percentage link changes per iteration for shapes 0dB image

2. Stop when the change in grey-level between iterations for every block has fallen below some negligible amount e.g 1.

In practice condition (1) is rarely achieved, although the percentage change does tend to 0 (figure 3.11). The second criterion is more realistic and in the experiments convergence is achieved in typically 10-20 iterations. These characteristics compare favourably with stochastic relaxation methods (e.g SA) [39] [95] [16] which often require hundreds of iterations. The behaviour is more in line with the Iterated Conditional Modes (ICM) method described by Besag [9] [10], which the process more closely resembles. In the experiments presented below, however, the process was left to run until there was no discernible change in the link states to demonstrate the long term behaviour.

Summary

The steps of the region estimation process are summarised in figure 3.12, where the input to the procedure is the lowpass pyramid of the image.

```
procedure Region_Estimation (Pyramid)

begin

    G = Node_Selection (Pyramid);

    while not Converged (G) do
    begin
        for each node e in G do
        begin
            Theta[e] = Orientation_Estimate (e);
            (r[e], a[e], b[e]) = Edge_Estimation (Theta[e], r[e])
        end

        Average_Edge_Estimates (G);

        for each node e in G do
        begin
            P(1) for all links L of e from
                    [Theta[e], r[e], a[e], b[e], P(edge)];
            Link decisions for L;
            Update x(e)
        end

        G' = Region_Refinement (G);
        G  = G';

    end

end
```

Figure 3.12: Region estimation algorithm

Procedure	*Storage (pixels)*		*Multiplications per pixel*	
Original Image	$N \times N$			
Pyramid	$\frac{4}{3}N^2$		4×4	
Node Selection			Inter-Level Var.	5
			Pyramid Smoothing	$(16 \times \frac{4}{3})$
Collapsing G	Nodes	$\frac{1}{20}N^2 \times 10$		
	Links	$\frac{5}{20}N^2 \times 3$		5
Iterative Estimation *(1 iteration)*			Edge Estimation	425
			Averaging	10
			Update	70
			Splitting	12
Total	$\frac{3}{2}N^2$		564	

Table 3.3: Computational requirements of region estimation

3.6 Computational Requirements

One of the features of the region segmentation scheme is its low storage and computational requirements. The storage requirements, in terms of pixels or processing sites, and the number of multiplications per pixel for each of the main procedures of the region process are shown in table 3.3. The initial number of nodes in the region graph G is clearly dependent on the image, so the statistics quoted in the table for the number of nodes and the average connectivity are taken from experimental results. Also as the regions are refined, a certain percentage of nodes are split into 4, increasing the number and the connectivity. For the calculations in the table, it has been assumed that the number of nodes in G is 5% of the number of pixels and that convergence is achieved after 20 iterations. Tables 3.4 and 3.5 show the initial and number of nodes and region links per level for the synthetic 'shapes 0dB' image and the natural 'girl' image (figure 1.1(a) and (b)). The average connectivity is given in the final column of each table.

From table 3.3 it can be seen that the storage requirement for the entire processing is about $\frac{3}{2}N^2$ and that approximately 564 multiplications per pixel are needed over 1 iteration. As there are only 5% of the 'pixels' from the original image in G then 564 multiplies per

level	*leaf nodes*	*No. links*	*Connectivity*
8	0	0	0
7	32	128	4
6	640	2573	4
5	320	1493	5
4	128	739	6
3	3	21	7
Total	1105	4954	5

Table 3.4: Initial nodes and connectivity per level for 'shapes' 0dB image

level	*leaf nodes*	*No. links*	*Connectivity*
8	0	0	0
7	5552	21882	4
6	1176	6372	5
5	271	1875	7
4	28	249	9
Total	7027	30378	4

Table 3.5: Initial nodes and connectivity per level for 'girl' image

pixel is the total requirement over 20 iterations. The major contribution to the computation is from the edge estimation, where the figure quoted (425 multiplies) has not been rationalised to take into account the repeated terms that appear in equations (3.28) through to (3.35). The results, however, compare favourably with the computation required for MAP estimation using SA, where many hundreds of iterations over the entire set of $N \times N$ pixels is often required [39]. The computation also compares well with that achieved by using ICM in a multigrid context, such as in [16], where the computational figures quoted are also about 2 orders of magnitude better than that possible using SA.

3.7 Results and Discussion

The results presented in this section illustrate the operation of the region estimation process. The synthetic 'shapes' image of figure 3.13 was specifically designed as a foreground/background test image with objects that have corners of varying acuteness and sides that are both curved and straight. The image is an 8-bit (256 grey levels), size

Figure 3.13: 'shapes' original

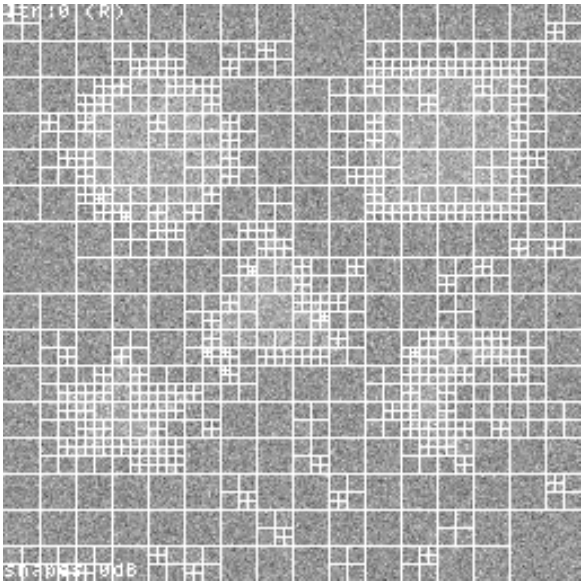
256×256 and the difference between the foreground and background is 20.

The region segmentation was applied to a noisy version of the 'shapes' image, with an inter-region signal to noise ratio of 0 dB. This is calculated as [88] [86] [87]

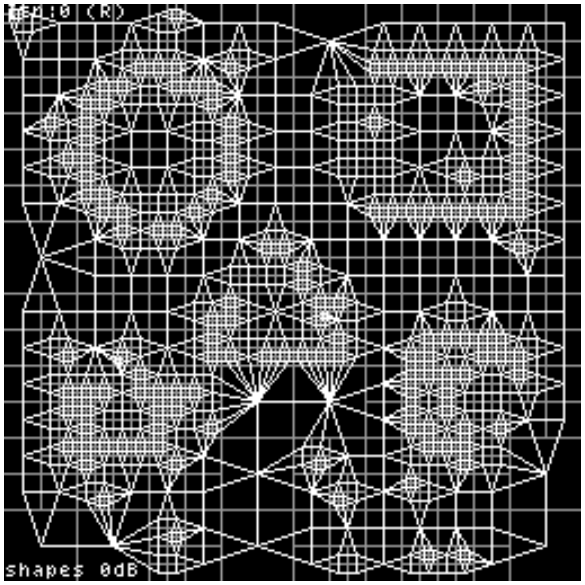
$$\text{Inter-region SNR} = 20 \log_{10} \left[\frac{\mu_1 - \mu_2}{\sigma_n} \right] \text{dB} \tag{3.50}$$

where μ_1 and μ_2 are the foreground and background grey levels ($\mu_1 = 170, \mu_2 = 150$ for the 'shapes' image), and σ_n is the s.d. of the additive Gaussian white noise. Figure 1.1(a) (in Chapter 1) shows the corrupted 'shapes' image SNR = 0dB ($\sigma_n = 20$) used for the following results.

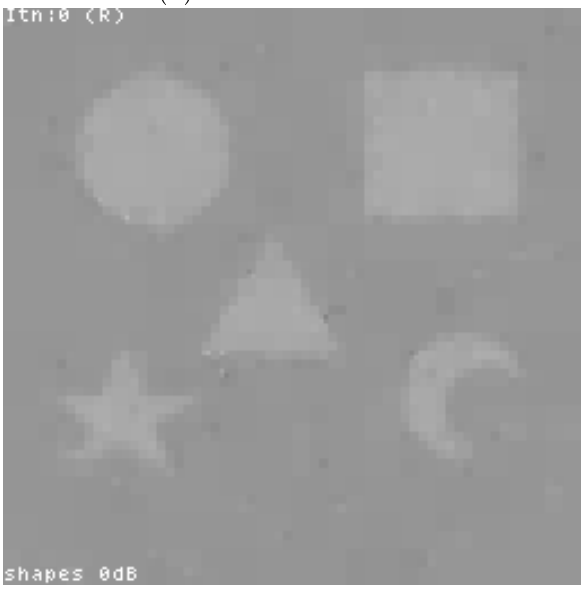
Figure 3.14(a) shows the result of the node selection stage. The input image has been overlayed with the boundaries of the image blocks represented by the selected nodes. The first thing to notice is that there is data reduction from the original 65536 pixels to about 1105 nodes (see table 3.4). The tessellation is not regular; large blocks represent nodes from lower spatial resolutions and smaller blocks, which generally cluster around the edges of the objects, are from higher spatial resolutions.



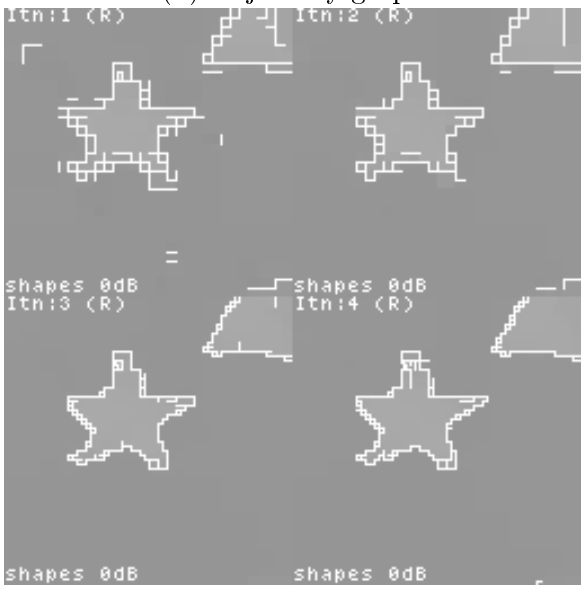
(a) Block tessellation



(b) Adjacency graph

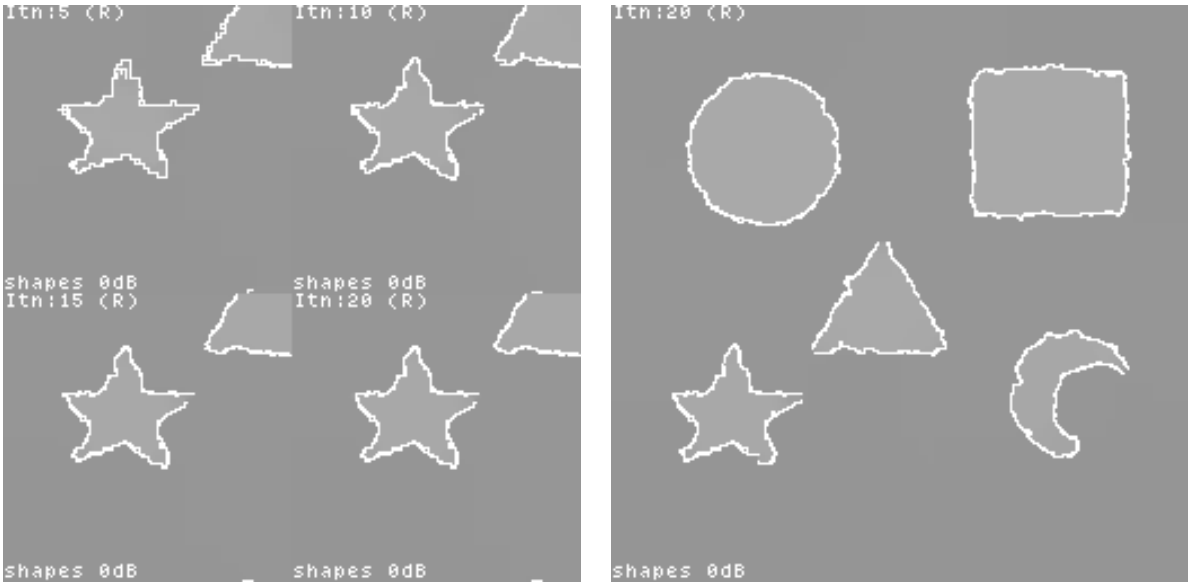


(c) Means



(d) Iterations 1 to 4

Figure 3.14: Region estimation of segmentation on 'shapes' 0dB



(a) Iterations 5 to 20 (b) Result after 20 iterations

Figure 3.15: Region estimation on 'shapes' 0dB cont.

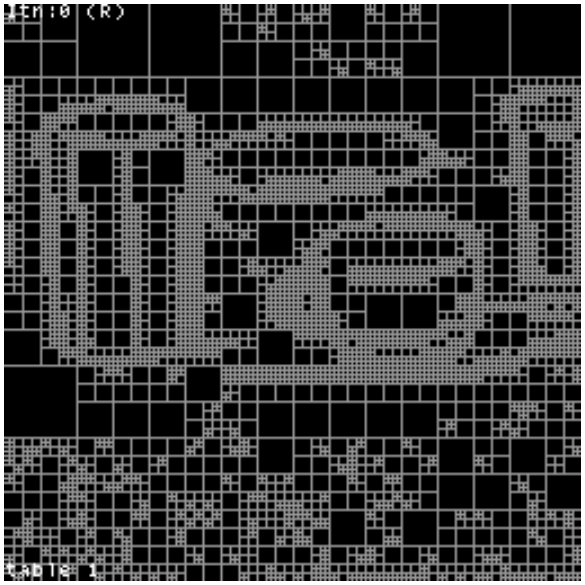
Figure 3.14(b) shows the region adjacency graph created from the inhomogeneous tessellation. The region links are drawn from the centres of the blocks. Note that this is only what the initial graph looks like. As the estimation proceeds certain nodes will be split into 4, and the new nodes (children) will themselves be linked to the existing neighbours. This will tend to occur at the edges of the objects as the edge blocks are successively refined. In figure 3.14(c) the current grey level data has been 'painted' into the representative region blocks, showing the initial region segmentation.

Figures 3.14(d) and 3.15(a) show only the bottom left hand quadrant of the image over several iterations. The figures show the evolution of the 'off' links drawn as white lines along the respective block edges. The region block refinement is apparent by the size of the white lines, which eventually end up being 1 or 2 pixels in width. The initial randomness of the link states rapidly vanishes as the edges of the objects are located and data is smoothed in the internal regions. The result after 20 iterations is shown in figure 3.15(b). All the objects have been successfully segmented and the boundary estimate is reasonable. The poor

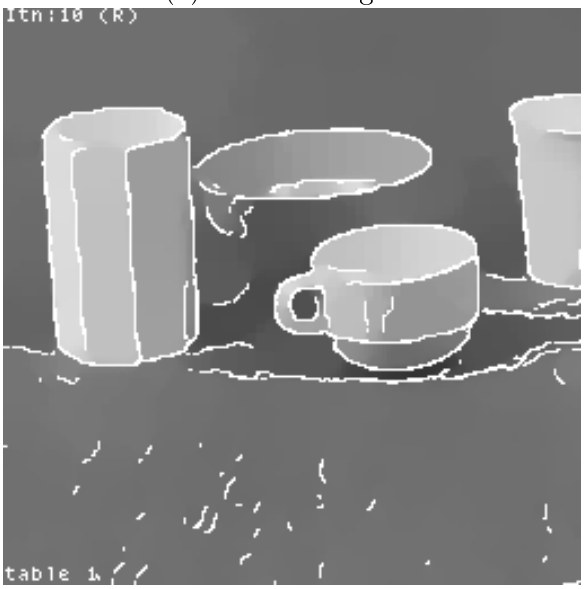
performance at the corners of the star and crescent is perhaps not surprising because of the simplicity of the boundary model used for the region orientation estimation (section 3.5.1).

Finally figures 3.16(a)-(c) show the results on the natural image 'table 1'. This image is also 256×256, 8-bit and has been chosen because it has many regions which are essentially 'flat', and also boundary shapes which are polygonal, which conforms better to the boundary model. The initial tessellation is shown in figure 3.16(b) and once again exhibits the characteristic clustering of small blocks around significant region boundaries. The result, shown in figure 3.16(c), is not as easy to interpret as that for the synthetic image. Although the major image regions have been identified, the local averaging has flattened out the regions. There is also poor performance on some of the more diffuse edges, such as the edges of the bowl in the background, which can be attributed to the inadequacies of the edge model.

(a) 'table 1' original



(b) Block Tessellation



(c) Result after 10 iteration

Figure 3.16: Region estimation on 'table 1' image

Chapter 4

Boundary Estimation Process

4.1 Introduction

This chapter details the boundary estimation process. There are two main sections in this chapter. First, the problem of robust estimation of orientation is addressed in the form of a multiresolution constrained MMSE estimation [102]. This an adaptation of the scale-space recursive filtering based on a linear multiresolution model of the image developed by Clippingdale [25]. Secondly a boundary segmentation process is described, which operates independently of the region estimation process, and uses an orientation pyramid.

4.2 Orientation Estimation

It was noted in Chapter 1 that edge information is of fundamental importance in both biological and computer vision systems and that this has been the motivation for its adoption by workers in image processing and computer vision. Too often, however, the essential parameter of the *orientation* of the edge, has been ignored. This is surprising when it has been shown that nerve cells called 'simple cells' exist in the human visual cortex which respond to linear features in the retinal image which have orientations lying within a narrow angular band [52] [32] [51]. In image segmentation, edge detection provides the necessary 'localisation' which is absent in purely region based approaches. And it is orientation which

provides the shape information which is relatively independent of the illumination.

Edge detection and estimation methods have received much attention since the early work of Roberts [81], leading to a wide variety of approaches. The major issues that affect the choice of a particular method are noise sensitivity, which is tied to the spatial localisation, and computational efficiency. The logical way to achieve noise immunity is by post processing of the orientation estimate by some form of 'smoothing'. The methods proposed to solve this problem range from local processing such as simple averaging [61], Bayesian estimation [39] or relaxation labelling [112], to what may be regarded as global processing such as the Hough Transform [34]. Much of the diversity of these methods stems from the variety of orientation representations used and in part from the underlying image models.

This section describes a robust orientation estimation method based on a multiresolution signal model [25] which is able to extract boundary orientations from data with 0dB SNR. The method is an extension of the scale-space recursive filtering developed by Clippingdale [26]. The method effectively uses the spatial consistency of outputs of small kernel gradient operators from different scales to produce more reliable edge position and orientation. The nearest counterpart to the work presented here is that of the original work by Rosenfeld and Thurston [83], Marr [69] and Witkin [110], who also combined the outputs of filters at multiple scales.

4.2.1 Orientation Representation

As mentioned above, the diversity of orientation estimation and subsequent smoothing methods is partly due to the varieties of orientation representation adopted. This section outlines the reasons for the choice in the present work of what is known as a *double angle* representation [61].

The representation of the orientation is particularly important if averaging is to be used,

which is generally the case if noise is present. Consider the commonly used *gradient vector* representation [41]. If the image locally consists of a single oriented feature, at angle θ

$$x_{ij} = f(i\cos\theta + j\sin\theta) = f(u) \tag{4.1}$$

then the gradient vector is simply

$$\mathbf{g}_{ij} = \left(\frac{\partial f}{\partial i}, \frac{\partial f}{\partial j}\right)^T = r(u)\begin{pmatrix}\cos\theta \\ \sin\theta\end{pmatrix} \tag{4.2}$$

where

$$r(u) = \frac{\partial f}{\partial u} \tag{4.3}$$

The first problem encountered with this representation is the sign ambiguity of the derivative $r(u)$; changing $r(u)$ to $-r(u)$ does not affect the orientation but changes the sign of $\mathbf{g}_{ij}$. Secondly if $f(u)$ is an even function then $\mathbf{g}_{ij} = \mathbf{0}$, i.e. no orientation estimate exists even if there is rapid variation in the neighbourhood of the image. These two problems are compounded if an average is required. If an average is calculated, say within some circle of radius R around the origin, then if the feature is an even function

$$\int_{-R}^{R}\int_{-\sqrt{R^2-i^2}}^{\sqrt{R^2-i^2}} \mathbf{g}_{ij}\, didj = \mathbf{0} \tag{4.4}$$

so that no meaningful average orientation can be found, even though there is a feature within the circle.

The *double angle* representation, originally proposed by Knutsson [61], solves the above problems. This representation can be related to the gradient representation by using a tensor formulation [12]. The tensor function G_{ij} is defined by

$$G_{ij} = \mathbf{g}_{ij}\mathbf{g}_{ij}^T = r_{ij}^2\begin{pmatrix}\cos^2\theta & \cos\theta\sin\theta \\ \cos\theta\sin\theta & \sin^2\theta\end{pmatrix} \tag{4.5}$$

There is no sign ambiguity in G_{ij} and the orientation is given by the principal eigenvector of G_{ij}

$$\mathbf{v}_{ij} = \begin{pmatrix}\cos\theta_{ij} \\ \sin\theta_{ij}\end{pmatrix} \tag{4.6}$$

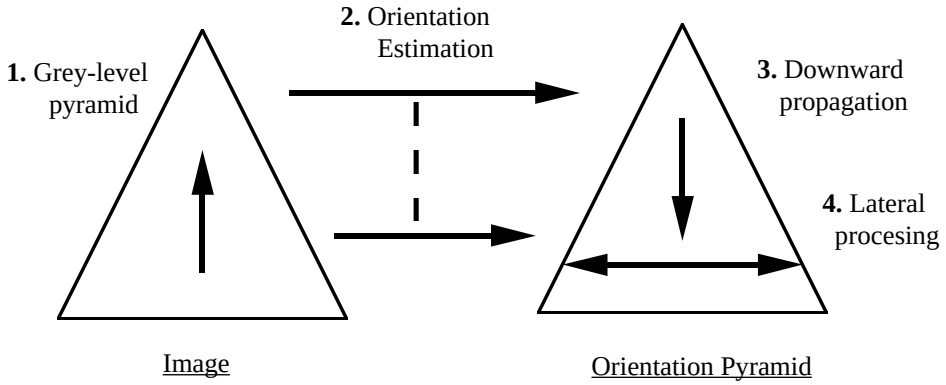


Figure 4.1: Orientation estimation algorithm structure

A less cumbersome formulation is an equivalent *double angle* vector representation

$$\boldsymbol{\theta}_{ij} = \begin{pmatrix} g_0^2 - g_1^2 \\ 2g_0g_1 \end{pmatrix} = r_{ij}^2 \begin{pmatrix} \cos 2\theta_{ij} \\ \sin 2\theta_{ij} \end{pmatrix} \tag{4.7}$$

Using this representation it can easily be verified that a meaningful average can be obtained.

Finally it is perhaps worth mentioning the case for using a *continuous* orientation representation, such as used by Granlund *et al* [42] [61], as opposed to a quantised or *discrete* form, as advocated by Zucker [112]. The main argument against the latter is that it is undesirable to introduce quantisation errors on top of those already present as a result of noise in the estimate. There is also the fundamental drawback with a discrete representation that simple averaging is not possible. There are of course applications, such as coding [93], where a quantised form is convenient.

4.2.2 An Orientation Pyramid

There are three steps to the estimation algorithm: construction of a grey level pyramid, orientation estimation, and smoothing of the initial estimates by scale-space recursion in the pyramid. The process is illustrated in figure 4.1. The scale-space recursion is a downward propagation of the more reliable estimates from higher levels of the pyramid to lower levels (greater spatial resolution), to reduce estimation errors.

-0.074	-0.095	0.000
-0.095	0.000	0.095
0.000	0.095	0.074

0.000	0.095	0.074
-0.095	0.000	0.095
-0.074	-0.095	0.000

Table 4.1: Optimised 3×3 edge kernels g_0 and g_1

The low-pass filter kernels used in the construction of the grey level pyramid were specifically designed to be circularly symmetric, so as not to introduce bias at this stage of processing. The initial orientation estimation is obtained by the use of a pair of optimised edge kernels, designed to minimise orientation bias and shown to perform better than other comparable small size edge kernels (a full account of the design methodology and performance is given in Appendix B). Table 4.1 shows the odd sized, 3×3 edge masks (g_0 and g_1) used. The estimate $\tilde{\boldsymbol{\theta}}_{ij}$ is obtained by combining the outputs of the spatial convolution using the double angle representation of equation (4.7)

$$\tilde{\boldsymbol{\theta}}_{ij} = \begin{pmatrix} g^2_{0\,(ij)} - g^2_{1\,(ij)} \\ 2g_{0\,(ij)}g_{1\,(ij)} \end{pmatrix} \tag{4.8}$$

The magnitude of this vector $|\tilde{\boldsymbol{\theta}}_{ij}|$ represents the local edge energy and the argument $\arg(\tilde{\boldsymbol{\theta}}_{ij})$ is *twice* the local orientation (with a rotation of $\pi/4$ because the edge masks are in fact oriented at $-\pi/4$ and $+\pi/4$). The estimation is performed on each level of the grey level pyramid creating an *orientation pyramid* representation of the image.

4.2.3 Vertical Propagation

As already noted, the orientation estimates at higher levels of the pyramid, where the noise has been reduced by pyramid smoothing, will be more reliable than those at the lower levels. The trade-off, however, is that at the higher levels the spatial resolution is low, i.e there is not the required localisation, which is an aim of edge detection. Under the assumption that features exist over more than one level of the orientation pyramid, what may be termed as the assumption of *scale consistency* [110] [100], it is to be hoped that by propagating down the estimates from the more reliable data, estimation errors may be reduced at the

lower levels and the necessary spatial resolution recovered. The effect of averaging *between* levels is that consistent features are emphasised, while at the same time noise, which will be uncorrelated between levels, will be averaged out.

A Linear Multiresolution Signal Model

The variation of orientation across the image is modelled as an *evolution* through *scale* via a linear equation of the form (section 2.3.1)

$$\boldsymbol{\theta}_{ij}(l) = \sum_{m=-K}^{K} \sum_{n=-K}^{K} B_{mn} \boldsymbol{\theta}_{i/2+m,j/2+n}(l-1) + \boldsymbol{\nu}_{ij}(l) \tag{4.9}$$

where B_{mn} is a scalar interpolation filter and $\boldsymbol{\nu}_{ij}(l)$ an *innovation* vector. This model is a vector form of the scalar linear multiresolution signal models described in [26] [25]. Beginning with the root of the pyramid, vectors on each successive level are modelled as a propagated term *plus* an innovation term. This effects a refinement process through scale-space and encapsulates the assumption of consistency of features over several levels.

The simplest form of this model is defined on a quadtree, with

$$\boldsymbol{\theta}_{ij}(l) = \boldsymbol{\theta}_{i/2,j/2}(l-1) + \boldsymbol{\nu}_{ij}(l) \tag{4.10}$$

In this case it can be shown that the MMSE estimator $\hat{\boldsymbol{\theta}}_{ij}(l)$, given the set of noisy data

$$\tilde{\boldsymbol{\theta}}_{ij}(l) = \boldsymbol{\theta}_{ij}(l) + \boldsymbol{\eta}_{ij}(l) \tag{4.11}$$

is just

$$\begin{aligned} \hat{\boldsymbol{\theta}}_{ij}(l) &= \beta_{ij}(l)\hat{\boldsymbol{\theta}}_{i/2,j/2}(l-1) + (1-\beta_{ij}(l))\tilde{\boldsymbol{\theta}}_{ij}(l) \\ \hat{\boldsymbol{\theta}}(0) &= \tilde{\boldsymbol{\theta}}(0) \end{aligned} \tag{4.12}$$

where the *feedback* coefficients $\beta_{ij}(l)$ are a function of the signal and noise variances [25]. Thus the best estimator of $\boldsymbol{\theta}(l)$ can be expressed as a first order, recursive filter operating in scale-space.

Now comparing equation (4.9) and equation (4.12) suggest a more general form of a sub-optimal estimator

$$\hat{\boldsymbol{\theta}}_{ij}(l) = \beta_{ij}(l) \sum_{m=-K}^{K} \sum_{n=-K}^{K} B_{mn} \hat{\boldsymbol{\theta}}_{i/2+m,j/2+n}(l-1) + (1-\beta_{ij}(l))\tilde{\boldsymbol{\theta}}_{ij}(l) \tag{4.13}$$

The inclusion of the interpolation filter clearly makes the estimator more computationally expensive than the quadtree estimator. However, if B_{mn} is kept small then the price is still less than an equivalent image filter. In fact the same 4×4 low-pass kernel used to construct the grey level pyramid was used as the interpolation filter in the experiments presented below.

4.2.4 Parameter Estimation

The feedback coefficients $\beta(l)$ control the amount of information that is propagated from the higher levels of the orientation pyramid and combined with the less reliable estimates at the lower levels. For noisy orientation data, $\beta(l)$ will decrease as $l \to M$. Given the signal and the data

$$\boldsymbol{\theta} = \begin{pmatrix} \theta_0 \\ \theta_1 \end{pmatrix}, \tilde{\boldsymbol{\theta}} = \begin{pmatrix} \tilde{\theta}_0 \\ \tilde{\theta}_1 \end{pmatrix} \tag{4.14}$$

the linear form of the estimator is given by

$$\hat{\boldsymbol{\theta}} = \mathbf{A} \begin{pmatrix} \tilde{\theta}_0 \\ \tilde{\theta}_1 \end{pmatrix} = \begin{pmatrix} \sum_q a_{0q}\tilde{\theta}_q \\ \sum_q a_{1q}\tilde{\theta}_q \end{pmatrix}, \; q = 0, 1 \tag{4.15}$$

where the subscripts to scalar variables in this section p, q denote vector components rather than position coordinates. The orthogonality principle [75] states that the errors should be orthogonal to all the data

$$E[(\theta_q - \hat{\theta}_q)\tilde{\theta}_p] = 0, \; p, q = 0, 1 \tag{4.16}$$

which is a set of scalar equations for all combinations of p and q. In practice, however, it can be assumed that the components in perpendicular directions are uncorrelated i.e. that

θ_q and $\tilde{\theta}_p$ are independent if $q \neq p$, giving the equations

$$E[\theta_q \tilde{\theta}_q] = E[\hat{\theta}_q \tilde{\theta}_q],\ q = 0, 1, \tag{4.17}$$

Assuming that the signal and noise are uncorrelated across and between levels of the orientation pyramid, then from equations (4.11), (4.13) and (4.17), for level l

$$\begin{aligned} E[\theta_q(l)\tilde{\theta}_q(l)] &= E[\hat{\theta}_q(l)\tilde{\theta}_q(l)] \\ \Rightarrow E[\theta_q(l)(\theta_q(l) + \eta_q(l))] &= E[\Big(\beta(l)\hat{\theta}_q(l-1) + (1-\beta(l))\tilde{\theta}_q(l)\Big)\,\tilde{\theta}_q(l)], \\ \Rightarrow E[\theta_q^2(l)] = \beta(l)E[\hat{\theta}_q(l-1)\tilde{\theta}_q(l)] &+ (1-\beta(l))E[\tilde{\theta}_q^2(l)],\ q = 0, 1 \end{aligned} \tag{4.18}$$

since $E[\theta_q(l)\eta_q(l)] = 0,\ q = 0, 1$. Also the errors at the current level are orthogonal to the estimate from the parent level, thus

$$\begin{aligned} E[\theta_q(l)\hat{\theta}_q(l-1)] &= E[\hat{\theta}_q(l)\hat{\theta}_q(l-1)],\ q = 0, 1 \\ \Rightarrow E[\theta_q(l)\hat{\theta}_q(l-1)] &= \beta(l)E[\hat{\theta}_q^2(l-1)] + (1-\beta(l))E[\tilde{\theta}_q(l)\hat{\theta}_q(l-1)] \end{aligned} \tag{4.19}$$

Considering only 1 component ($q = 0$), the simultaneous equations (4.18) and (4.19) may be expressed in matrix form

$$\boldsymbol{\alpha} = \mathbf{C}\boldsymbol{\beta} \tag{4.20}$$

where the feedback coefficients are

$$\boldsymbol{\beta} = \begin{pmatrix} \beta(l) \\ 1-\beta(l) \end{pmatrix} \tag{4.21}$$

and the matrix is

$$\mathbf{C} = \begin{pmatrix} E[\hat{\theta}(l-1)\tilde{\theta}(l)] & E[\tilde{\theta}^2(l)] \\ E[\hat{\theta}^2(l-1)] & E[\tilde{\theta}(l)\hat{\theta}(l-1)] \end{pmatrix} \tag{4.22}$$

Using equation (4.11) to determine the $\boldsymbol{\alpha}$ i.e.

$$E[\theta^2(l)] = E[\tilde{\theta}^2(l)] - E[\eta^2(l)] \tag{4.23}$$

$E[\hat{\theta}^2(l-1)]$	Estimate of variance of parent level
$E[\hat{\theta}^2(l)]$	Estimate of variance of child level
$E[\tilde{\theta}(l)\hat{\theta}(l-1)]$	Covariance of data on parent and child levels
$E[\eta^2(l)]$	Estimate of noise variance on child level

Table 4.2: Correlation statistics required to find feedback coefficients

and

$$\begin{aligned} E[\theta(l)\hat{\theta}(l-1)] &= E[(\tilde{\theta}(l)-\eta(l))\hat{\theta}(l-1)] \\ &= E[\tilde{\theta}(l)\hat{\theta}(l-1)] \end{aligned} \tag{4.24}$$

thus

$$\boldsymbol{\alpha} = \begin{pmatrix} E[\tilde{\theta}^2(l)] - E[\eta^2(l)] \\ E[\tilde{\theta}(l)\hat{\theta}(l-1)] \end{pmatrix} \tag{4.25}$$

Inverting the matrix $\mathbf{C}$ in equation (4.20) then yields the feedback coefficients $\boldsymbol{\beta}$. The correlation statistics required to determine these coefficients are summarised in table 4.2.

As with the region estimation process (section 3.3.1) all parameter estimation takes place within the pyramid structure. Furthermore 'local' estimates for each of the correlation statistics listed in table 4.2 are obtained by determining averages using pyramid smoothing (figure 3.2).

The question of estimating the noise variance arises at this point. One simple, but effective, method is to threshold the magnitudes of the orientation vectors at some level $l < M$ and then to use the resulting edge/non-edge classification at all levels below and including l to estimate the noise across the (assumed) constant regions, free from influence of the signal. The choice of this 'highest' level, however, is more problematical. Because the edge kernels are have a finite size there is a lowest spatial resolution above which the orientation estimates cannot be made because the image is too small. For an edge kernel of size say 3×3 there can only be 4 correct estimates on the pyramid level size 4×4, so this becomes the limiting level. For quite high levels of noise (SNR $<$ 0dB) sufficient

smoothing is achieved at the level of size 32×32, so this would seem like a suitable level at which to start the recursive estimation. When the orientation estimation is to be used for the segmentation process outlined in Chapter 3, then the highest level of region process tessellation is the automatic choice for scale l.

Choosing a level above the 'optimal' highest level does not adversely affect the vertical propagation as the feedback coefficients moderate the amount of energy passed down until the estimated noise becomes significant. The drawback with thresholding to determine an estimate of noise, however, is that fine structures, which are smoothed out at the higher levels of the pyramid, will be measured as noise.

4.2.5 Lateral Smoothing

The scale-space recursion of equation (4.13) can be supplemented by iterated *lateral* processing which produces an improvement in the estimate. The propagated orientation estimate $\hat{\boldsymbol{\theta}}(l)$ is anisotropically filtered at each level by a 2-d Gaussian kernel $H(\theta)$. The iterated filtering takes the form

$$\hat{\boldsymbol{\theta}}^{t}_{ij}(l) = \sum_{m}\sum_{n} H_{mn}\left(\hat{\theta}^{t-1}_{ij}(l)\right)\hat{\boldsymbol{\theta}}^{t-1}_{i+m,j+n}(l) \tag{4.26}$$

The smoothing kernel $H_{mn}(\theta)$ is oriented depending on the last estimated orientation

$$\hat{\theta}^{t}_{ij}(l) = 0.5\,\arg\left(\hat{\boldsymbol{\theta}}^{t}_{ij}(l)\right) \tag{4.27}$$

and $\hat{\boldsymbol{\theta}}^{0}(l)$ is given by equation (4.13). The filter kernels can be implemented by interpolation from a fixed set of kernels as in [61]. The method adopted here, however, is more elaborate: the filter shape is determined by constrained optimisation.

The 2-d Gaussian filter may be separated into 1-d functions along and perpendicular to the (major) axis of the filter as illustrated in figure 4.2.

$$H_{mn}(\theta) = H(x)H(y) \tag{4.28}$$

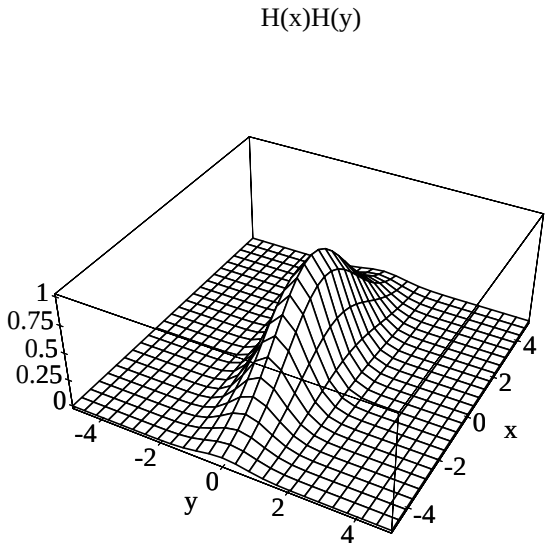


Figure 4.2: Anisotropic 2-d Gaussian filter

where

$$\begin{aligned} x &= m\cos\theta - n\sin\theta \\ y &= m\sin\theta + n\cos\theta \end{aligned} \tag{4.29}$$

and

$$\begin{aligned} H(x) &= \alpha_x \exp(-\beta_x x^2) \\ H(y) &= \alpha_y \exp(-\beta_y y^2) \end{aligned} \tag{4.30}$$

The correlation function of the orientation data is approximately separable in the orientation of the feature and Gaussian in shape. This is because it is primarily the result of the pyramid which itself is an approximation to Gaussian filtering. Thus the signal correlation

can also be separated along and perpendicular to the feature orientation θ

$$R_{\theta\theta} = R_{\theta\theta}(x)R_{\theta\theta}(y) \tag{4.31}$$

where

$$\begin{aligned} R_{\theta\theta}(x) &= \exp(-\rho_x x^2) \\ R_{\theta\theta}(y) &= \exp(-\rho_y y^2) \end{aligned} \tag{4.32}$$

And similarly the noise correlation

$$\begin{aligned} R_{\eta\eta}(x) &= \eta\delta(x) \\ R_{\eta\eta}(y) &= \eta\delta(y) \end{aligned} \tag{4.33}$$

The orientation energy will be highly correlated along the direction of the feature ($\vec{x}$), with the spread in the perpendicular direction ($\vec{y}$) being greater at lower levels of the pyramid because of the noise.

To determine the shape of the filter, the problem is posed as a least squares minimisation. Because both the filter and the correlation are separable the problem reduces to the 1-d minimisation. It can be shown that the constrained 1-d least squares solution for a Gaussian filter

$$h(x) = \alpha \exp(-\beta x^2) \tag{4.34}$$

given that the correlation functions for the signal and noise are

$$\begin{aligned} R_\theta(x) &= \exp(-\rho x^2) \\ R_n(x) &= \sigma\delta(x) \end{aligned} \tag{4.35}$$

has parameters

$$\alpha = \sqrt{\frac{2\beta(\beta+2\rho)}{(\beta+\rho)}}\,\frac{1}{(\sqrt{2\pi}+\sigma\sqrt{\beta+2\rho})} \tag{4.36}$$

$$\beta = \rho + \frac{2\rho^2\sqrt{2\pi}}{\sigma(\beta + 2\rho)^{\frac{3}{2}}} \tag{4.37}$$

The full derivation is detailed in Appendix A. It is possible to determine β numerically, for example using the Bisection Method [50], and hence α can be found. The parameter of the 2-d Gaussian correlation function $[\rho_x, \rho_y]$ can be found by least squares curve fitting [78], to the correlation data gathered from the orientation estimate on the current level. The next section describes how the correlation data is gathered.

4.2.6 Estimating Signal Correlation

The correlation data is gathered by sampling the orientation data at each point in a quadrant of a circle $[0 \leq r \leq R, \theta - \pi/2 \leq \phi \leq \theta]$ defined around the estimate $\boldsymbol{\theta}_{ij}$. The data is sampled in polar form at a discrete set of orientations $\triangle\phi$ and radii $\triangle r$, see figure 4.3. The data at the sample points (r, ϕ) is obtained by a bilinear interpolation from the 4 neighbours. The correlation function is thus built up by

$$R_{\theta\theta}(r, \phi) = \sum_{ij} \boldsymbol{\theta}_{ij}^T \boldsymbol{\theta}_{r,\phi},\ 0 \leq r \leq R,\ \theta - \pi/2 \leq \phi \leq \theta \tag{4.38}$$

The parameters $[\rho_x, \rho_y]$ are then determined by least squares fitting a 2-d Gaussian function to the gathered data.

Anisotropic Filtering

Having determined the parameters of the Gaussian filter $H(\theta)$, namely $[\alpha_x, \beta_x, \alpha_y, \beta_y]$, the filtering is performed by iterated spatial convolution using a set of n oriented filters

$$H(\theta_k),\ \theta_k = \frac{k\pi}{n-1}, 0 \leq k < n \tag{4.39}$$

The appropriate filter is selected at each point using $\hat{\theta}^t$ from equation (4.27). The number of iterations of this filter I, is determined by the maximum of the two standard deviations

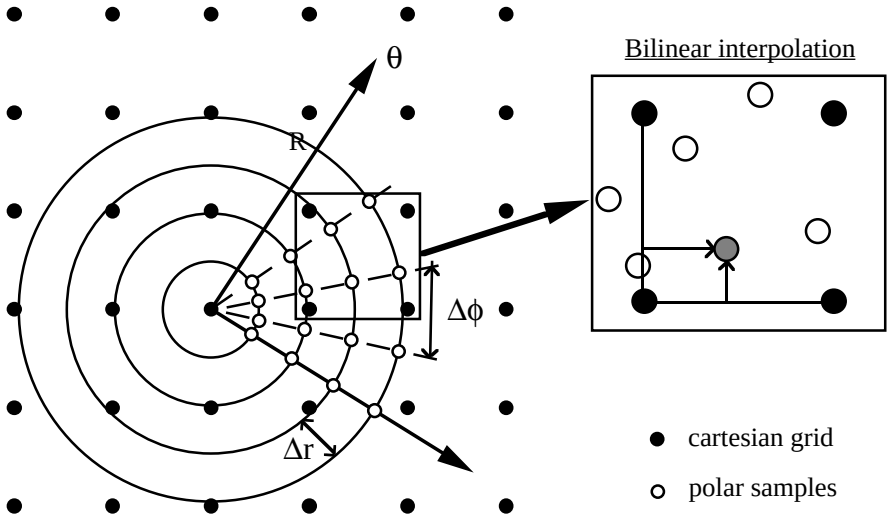


Figure 4.3: Gathering correlation data

of the filter

$$I = \left\lceil \left(\frac{3\sqrt{\max[\frac{1}{2\beta_x}, \frac{1}{2\beta_y}]}}{0.5K} \right)^2 \right\rceil \tag{4.40}$$

where K is the size of the spatial filter. In the following experiments a set of oriented filter kernels size 5×5 were used.

4.2.7 Computational Requirements

The computational requirements of the orientation estimation method described in this section are summarised in table 4.3. The per-pixel computation is $\frac{4}{3} \times 107 \approx 144$ multiplications per pixel, which is equivalent to a filter kernel of size 12×12. With the addition of the lateral processing, summarised in table 4.4, the total is increased to approximately $549 \approx 23 \times 23$ multiplications per pixel, thus the equivalent filter size is roughly quadrupled. The overhead lies in the correlation function estimation, which could be reduced by sampling over a smaller area. This, however, was not explored and it is hard to judge what the implications are to the fidelity of the result. The reason why the figures for the least squares fitting and the numerical approximation are small is that they are only performed once for each level of the pyramid so that there are only a few multiplications per pixel.

Procedure	*Storage (pixels)*	*Multiplications per pixel*	
Original Image	$N \times N$		
Pyramid	$\frac{4}{3}N^2$	4×4	
Orientation Estimation		22	
Parameter Estimation		Correlation stats.	8
		Feedback	4
		Smoothing	21
Vertical Propagation		36	
Total	$\frac{4}{3}N^2$	107	

Table 4.3: Computational requirements of orientation estimation

Procedure	*Storage (pixels)*	*Multiplications per pixel*	
Parameter Estimation	$\frac{4}{3}N^2$	Gathering correlation	200
		Least squares fitting	4
		Numerical solution	1
Iterative Convolution		100	
Total	$\frac{4}{3}N^2$	305	

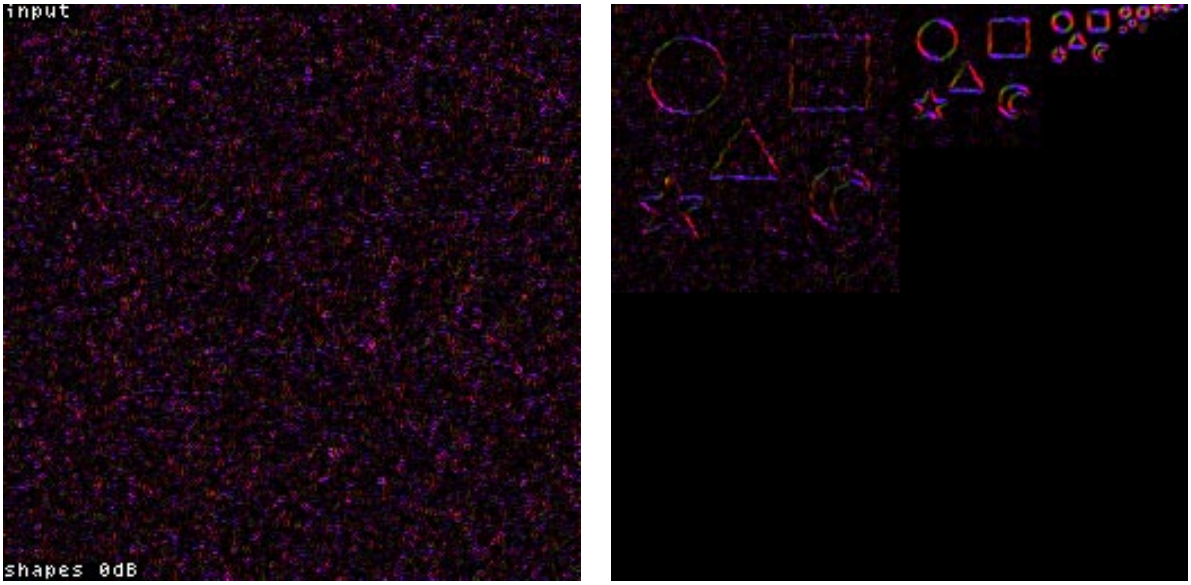
Table 4.4: Computation required for lateral smoothing

4.2.8 Results

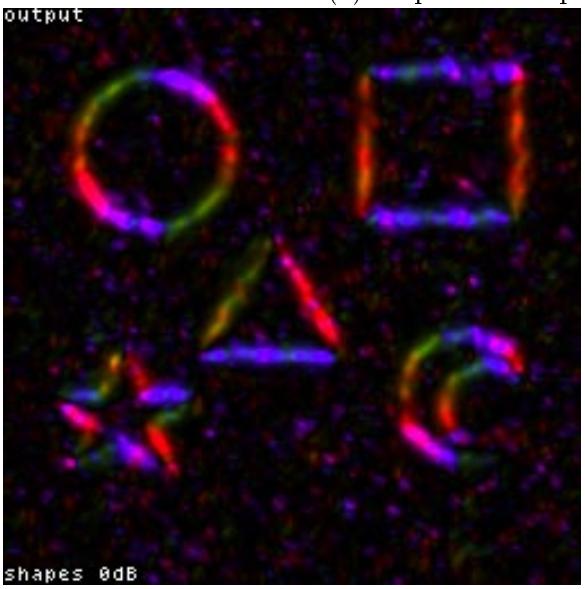
The estimation scheme was applied to a pyramid of the 'shapes' 0dB image (figure 1.1(a)). Figure 4.4 shows the input pyramid and the output of the processing. To obtain a measure of the SNR gain, the results were compared against an orientation pyramid of the noise-free 'shapes' image.

The input orientation pyramid (figure 4.4(a)) shows that there is practically no edge information on image plane level of the pyramid. Inspection of the lower spatial resolutions, however, shows the effect of the pyramid smoothing and the initial orientation estimation begins to pick up the edges at these levels. The result of propagation of these more reliable estimates down the pyramid is apparent in the output pyramid ((figure 4.4(b)). The gain in SNR is quantified in table 4.5.

Figure 4.5(a) shows a test image with 4 objects with irregular boundaries that have been generated by a random-walk process. To this noise has been added to give 4 SNRs

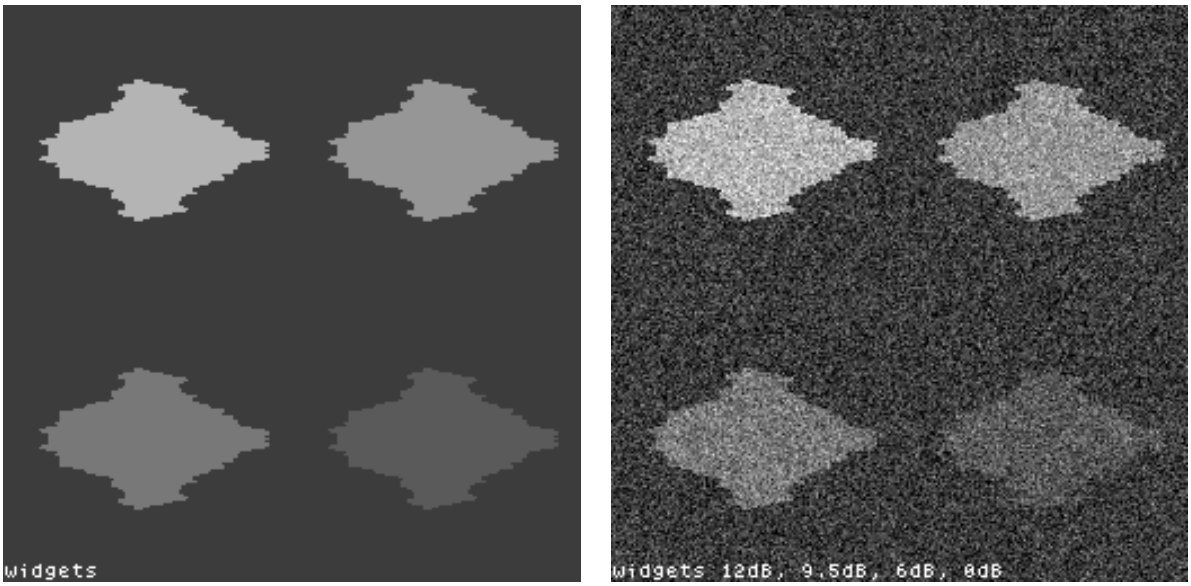


(a)‘shapes’ 0dB input orientation pyramid



(b)‘shapes’ 0dB output orientation pyramid

Figure 4.4: Orientation estimation on ‘shapes’ 0dB



(a) 'widgets' original (b) 'widgets' 12dB, 9.5dB, 6dB, 0dB

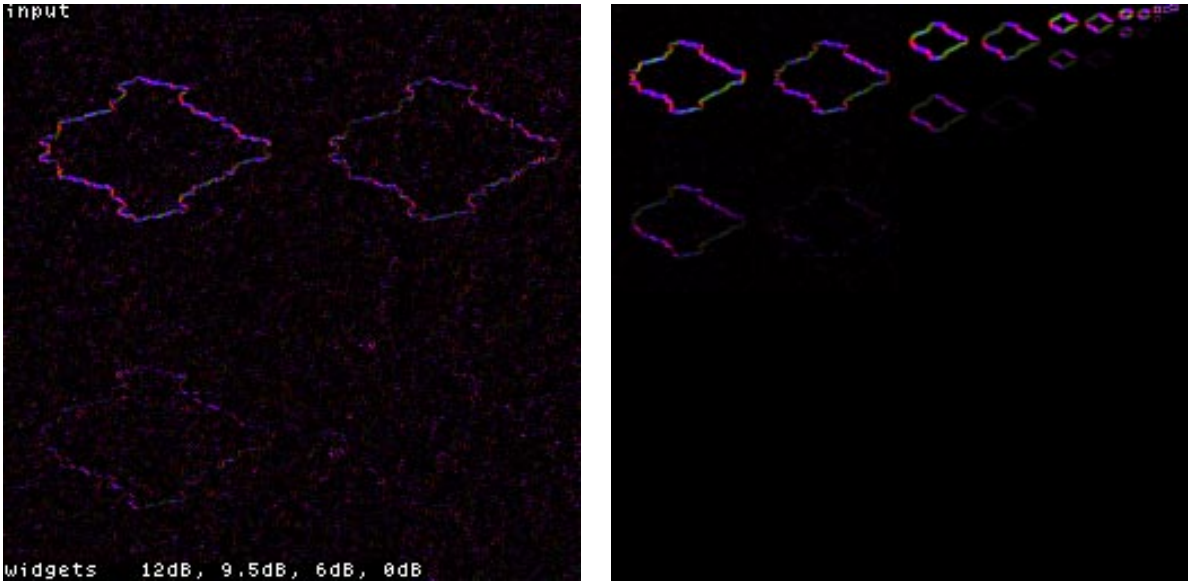
Figure 4.5: Irregular boundary synthetic image

of 12dB, 9.5dB, 6dB and 0dB in the same image. This image is important as it is able to test out the local statistic gathering ability of the estimation method. Figure 4.6 shows the input and output orientation pyramids. The input pyramid shows that the bottom right region (at 0dB) can only be detected about 2 levels above the image plane, whereas the top left region is visible on the image plane. The output demonstrates that the method has performed the smoothing adaptively.

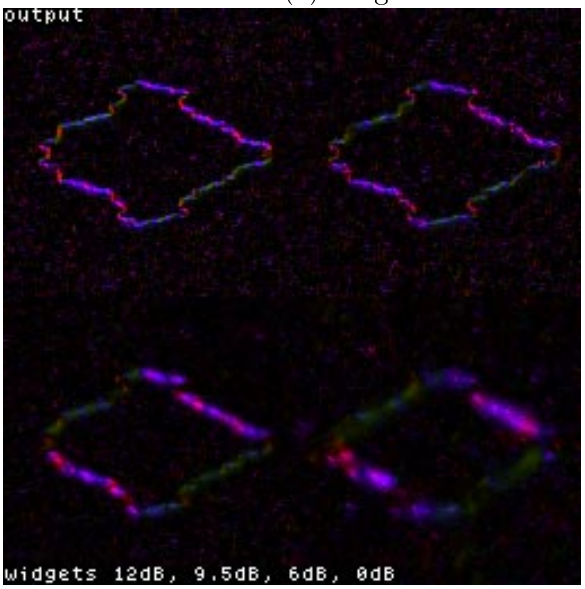
A SNR Measure

The signal to noise ratio measure is based on the correlation between the known orientation, obtained from an orientation estimate of a noise free version of the original image, with that of the restored estimate. The SNR is defined as

$$\mathbf{\Phi} = \frac{\sum_{ij} |\hat{\boldsymbol{\theta}}_{ij}(M)| \cos\Big(\arg(\hat{\boldsymbol{\theta}}_{ij}(M)) - \arg(\boldsymbol{\theta}_{ij}(M))\Big)}{\sum_{ij} |\hat{\boldsymbol{\theta}}_{ij}(M)|} \tag{4.41}$$



(a)'widgets' 12dB-0dB input orientation pyramid



(b)'widgets' 12dB-0dB output

Figure 4.6: Orientation estimation on 'widgets' 12dB-0dB

level	*Input SNR*	*Output SNR*
8	-4.4	23.4
7	13.1	31.1
6	27.5	33.9
5	38.4	40.4
4	50.9	50.9

Table 4.5: Input/Output SNRs per level for orientation pyramid of 'shapes' 0dB

The logarithmic SNR is defined to be

$$\text{SNR} = 10 \log_{10} \left[\frac{\mathbf{\Phi}}{\mathbf{1} - \mathbf{\Phi}} \right] \text{dB} \tag{4.42}$$

This measure is related to the angular error measure used in [62] and to the *squared vector* error used by Clippingdale [25]. It shares the property with the latter that the angular errors are weighted by the magnitude of the estimate $|\hat{\boldsymbol{\theta}}_{ij}(M)|$. This reflects the fact that in regions of the image where the grey level is more or less constant, the angular errors are less important than in the edge region, where the magnitude will be greater.

The angle estimation error can be defined as (the factor of a half is due to the double angle of the estimate)

$$\varepsilon_{\boldsymbol{\theta}} = 0.5 \cos^{-1}(\mathbf{\Phi}) \tag{4.43}$$

To give an idea of scale, a 1 degree error in angle corresponds to an SNR of about 32dB.

Table 4.5 shows the results of processing on levels of an orientation pyramid of the 'shapes' 0dB image, where the column marked *'Input SNR'* lists the SNRs before processing. The average angle estimation error $\varepsilon_{\boldsymbol{\theta}}$ is reduced from 33.4 degrees to 12.1. Table 4.6 summarises the input/output SNRs for various levels of noise added to the 'shapes' image, ranging from -6dB to 12dB. The data of table 4.6 is plotted in figure 4.7 together with results of orientation estimation without the lateral processing. The graph clearly shows that there is approximately 4dB gain over the vertical processing only.

Image SNR	*Input SNR*	*Output SNR*
-6dB	-18.5	16.1
-3dB	-11.4	20.5
0dB	-4.4	23.4
3dB	2.6	25.8
6dB	9.7	27.8
12dB	23.7	31.4

Table 4.6: Input/Output SNRs for various amounts of noise on 'shapes'

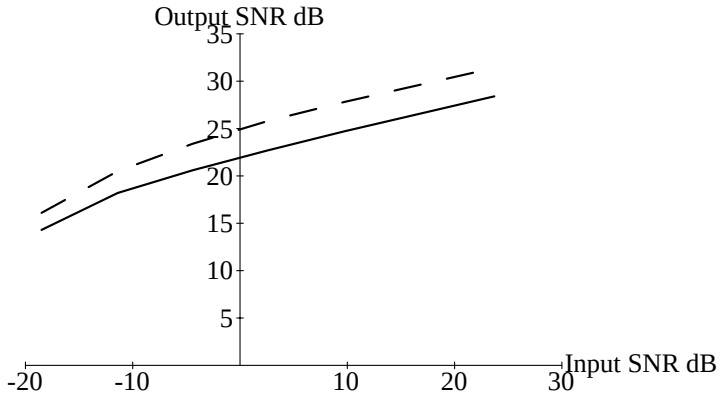


Figure 4.7: Effect of lateral processing (dashed line) on output SNR

4.3 Boundary Segmentation Process

The boundary estimation process is designed to operate independently of the region estimation process and assumes the segmentation model described in Chapter 2. The boundary process consists of 3 steps and, like the region process, is iterative. The overall paradigm is of least cost estimation using an 'optimal' representation in the form of a boundary graph, which is a dual of the RAG of the region process. This duality of the two process structures reflects the complementary nature of regions and their boundaries and provides the essential framework for making the two processes co-operate. The interaction between the processes to combine 'good' information from both, to achieve a better segmentation than

either process alone, is described in the Chapter 5.

The three main steps of the boundary process are: a node selection process that operates *top-down* on an orientation pyramid of the image, followed by a local linking process whereby neighbourhoods are created and an iterative relaxation process which allows the selected nodes to be moved to locally 'best' positions. The relaxation uses the orientation estimate as an energy landscape from which the best positions for the selected nodes are found by constrained search. The recovery of corners is achieved by this process.

4.3.1 Node Selection by Multiresolution Peak Detection

A set of boundary nodes is selected from the input orientation pyramid using a top-down, peak detection process. These nodes correspond to the vertex points of the polygonal boundaries of the image model.

1. An initial set of node candidates is selected from some pyramid level $l < M$. Similar arguments to those presented in section 4.2.4 regarding the selection of the start level for the orientation enhancement process can be made to select an appropriate resolution to begin the peak detection process. When the boundary process is run concurrently with the region estimation process, it is natural to begin at the highest level (lowest spatial resolution) automatically selected by the region block tessellation (section 3.3.2), as this indicates the largest region size. In the experimental results presented below the same level (32×32) as was used for the enhancement was adopted.

   The initial set of candidates are selected if their magnitudes exceed a threshold based on the mean local orientation magnitude. It is expected that at this resolution the noise variance will have been negligible, so a low threshold can be used to give the initial boundary set. As will become apparent, neither the start level nor this initial selection threshold is critical, as there is scope within the node selection procedure for

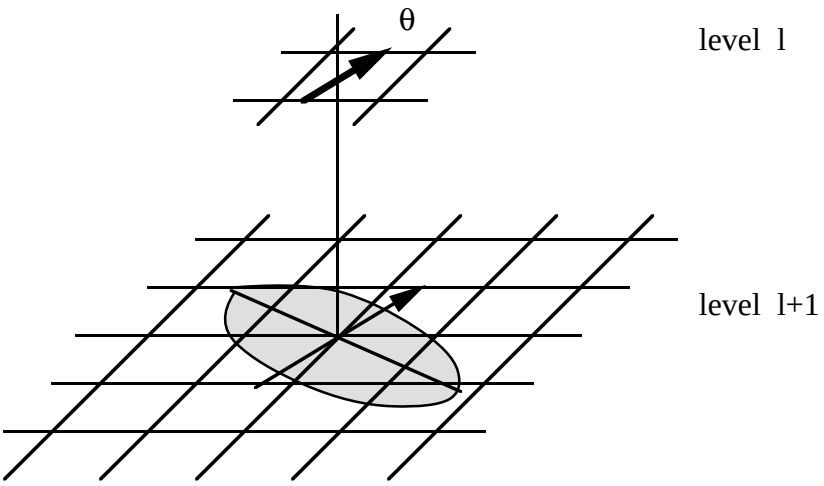


Figure 4.8: Peak detection search window

the addition of new nodes from levels below the start level.

2. A window is defined around each selected node, projected onto the next level of the pyramid, which has twice the resolution. The child nodes that lie within this window become the search area for the *peak detection* process, constrained by the orientation estimate of the parent, see figure 4.8. The simplest neighbourhood for the search is the 4 children as for a quadtree. This, however, leads to blocking, so an overlapping elliptical window was used whose major axis is oriented *perpendicular* to the orientation estimate of the parent. If (i, j, l) is a node at level l then the window is the set

$$X'(l+1) = \bigcup_{m,n \in E(a,b,\theta')} (2i+m, 2j+n, l+1) \tag{4.44}$$

where $E(a, b, \theta')$ represents an ellipse of half major axis a and half minor axis b, oriented at $\theta' = 0.5\arg(\boldsymbol{\theta}_{ij}(l)) + \frac{\pi}{2}$.

Next a child is selected within this window that has the maximum inner product with the parent (i, j, l). Select $(p, q, l+1) \in X'(l+1)$ which gives the maximum inner

product i.e.

$$\max \left[\bigcup_{(m,n)\in X'(l+1)} |\boldsymbol{\theta}_{mn}(l+1)||\boldsymbol{\theta}_{ij}(l)| \cos\left(0.5\arg(\boldsymbol{\theta}_{mn}(l+1)) - \arg(\boldsymbol{\theta}_{ij}(l))\right) \right] \quad (4.45)$$

Finally this set is augmented with any nodes in the complement area to the tested nodes in equation (4.45) (i.e. that are not in the window of any parent), but have energy which is significantly greater than the estimate of the noise variance at that level.

3. Step 2 is repeated at the next level of the orientation pyramid using the child nodes as the new parent set. The process thus proceeds recursively until the image plane is reached. If a selected node $\boldsymbol{\theta}_{ij}(l)$ is denoted by the scale-space position vector $\vec{\chi} = (i, j, l)^T$, then the initial set of n selected nodes forms the set $X = \{\vec{\chi}_0, \vec{\chi}_1, \ldots, \vec{\chi}_{n-1}\}$.

The node selection is robust because the peak detection search is constrained by the information from higher levels. It again utilises the better class information in the form of a boundary/non-boundary classification at the higher levels of the pyramid and propagates this down through the windowed node selection to regain spatial resolution. This process is related to the centroid methods described in [108] and has similarities to edge focusing [7] and other multiresolution edge detection methods [5].

4.3.2 Node Linking to form a Boundary Graph

The result of the above boundary node selection provides no connectivity information. There have been many heuristic rules designed to achieve the linking of edge points to form more or less continuous boundaries (see [5] for a good review). The main parameters that can be used to link together edge points to form curves are: distance, orientation and curvature.

The linking process is kept local, with each node $\vec{\chi}$ allowed to produce 2 links, denoted by $\lambda(\vec{\chi}, \vec{\psi})$, which are chosen from candidates in an elliptical neighbourhood. The size of

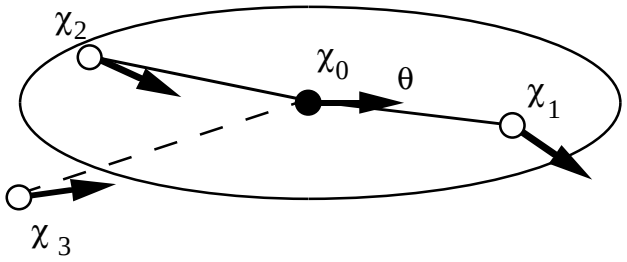


Figure 4.9: Boundary node linking

the window is governed by the level at which the node originally selected. A natural choice for the major axis of the window is 2^l oriented *along* the estimate at the node, from which the two best opposite links are chosen, see figure 4.9. The 'cost' of a link is calculated as the average orientation energy under the link, projected in the direction of the link

$$C(\lambda(\vec{\chi},\vec{\psi})) = \frac{1}{\mathrm{card}\{\Lambda'_{\vec{\chi},\vec{\psi}}\}} \sum_{m,n\in\Lambda'_{\vec{\xi},\vec{\psi}}} |\boldsymbol{\theta}_{mn}(M)| \cos\left(0.5\,\arg(\boldsymbol{\theta}_{mn}(M)) - \arg(\lambda(\vec{\chi},\vec{\psi}))\right) \quad (4.46)$$

where $\Lambda'_{\vec{\chi},\vec{\psi}}$ denotes the set of pixels that lie 'under' the link. In practice the data is sampled along the link and bilinear interpolation is used to determine the orientation at any given point.

This is an important cost function which is used in subsequent boundary relaxation. Note that by this linking method it is possible to have nodes with degree more than 2 as each node is initially allowed to have 2 links and after the linking each link is made bi-directional. Thus each node $\vec{\chi}$ has a set of neighbours $\mathcal{N}_{\vec{\chi}}$, and a set of links in this neighbourhood can be defined as all legitimate pairs

$$\mathcal{L}_{\vec{\chi}} = \bigcup_{\vec{\psi}\in\mathcal{N}_{\vec{\chi}}} \lambda(\vec{\chi},\vec{\psi}) \quad (4.47)$$

The complete set of links is then the set $L = \mathcal{L}_{\vec{\chi}_0} \cup \mathcal{L}_{\vec{\chi}_1} \cup \ldots \cup \mathcal{L}_{\vec{\chi}_{n-1}}$. The resulting structure $B = \{X, L\}$ is a Boundary Graph and, in some sense, forms a dual of the RAG of the region

estimation process. Figure 4.10 illustrates an example of the node selection and shows the resulting boundary graph.

4.3.3 Iterative Boundary Refinement

The graph created by the node selection and linking is the starting point of the relaxation process. The aim is to use the information in the neighbourhood of each node to move it to a better position. A cost is attached to each node evaluated at its current position based only on its immediate neighbours. Then by a constrained search around the current position, a position that has a lower cost is found.

There are 2 steps that control the movement of the nodes in the relaxation process at each node in the boundary graph. There are 2 further steps that control the elimination and introduction of nodes, which is related to the goodness of fit of the representation of the underlying curve to the local graph segment. These steps are performed at each node and repeated until there is no significant change between iterations. They may be summarised as:

1. Constrained search that aims to improve a local, anisotropic cost component.
2. A constrained search based on a cost component which is isotropic and inversely proportional to the distance between nodes.
3. A node elimination step based on a ML decision about the likelihood of the orientation data given the node.
4. A node addition step also based on the degree of fit with the orientation data.

1. Constrained Search: Anisotropic Cost Component

Each node is allowed to move either perpendicular or parallel to the direction of the local orientation which is estimated by taking the average orientation vector along the links from

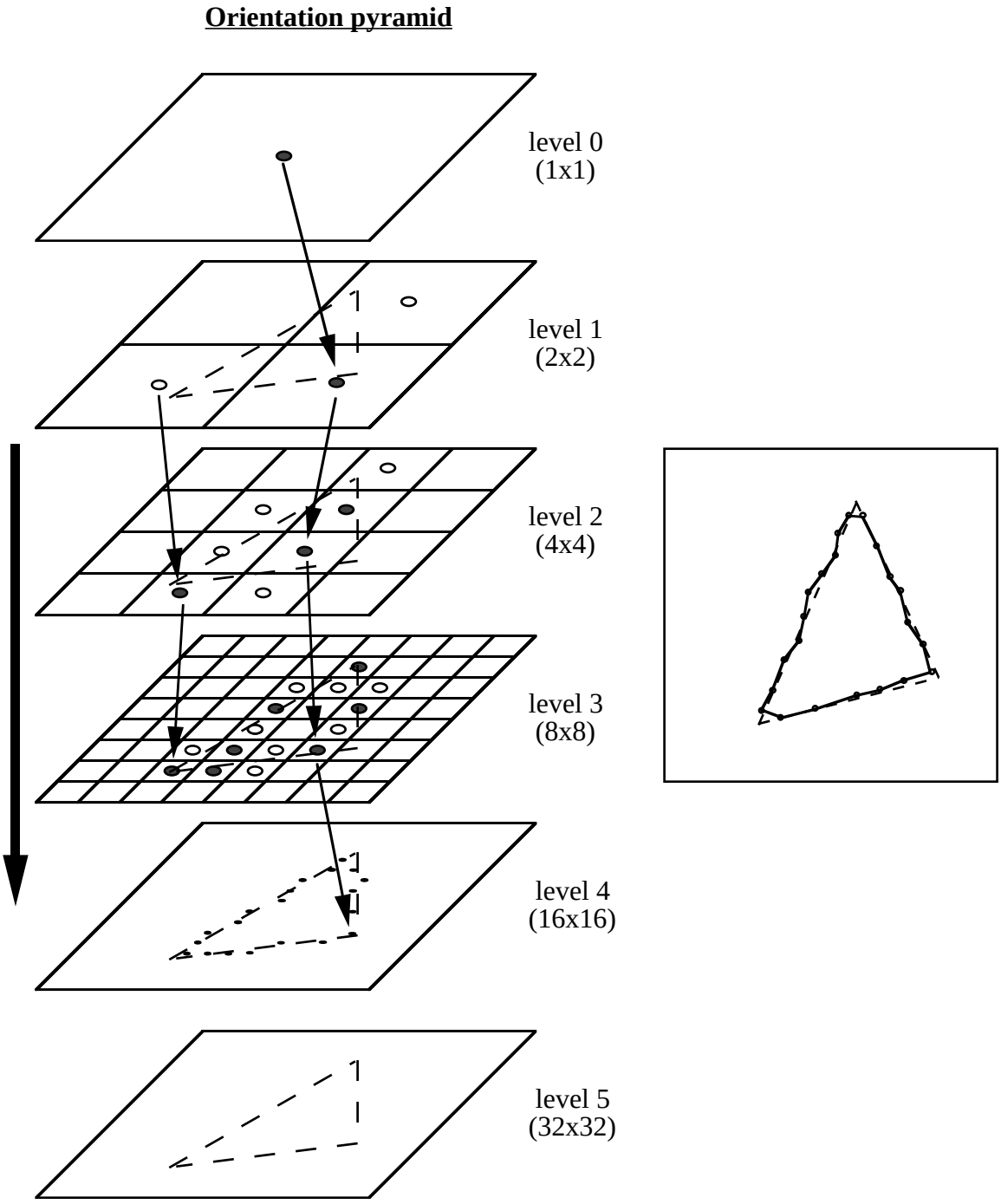


Figure 4.10: Boundary node selection and boundary graph

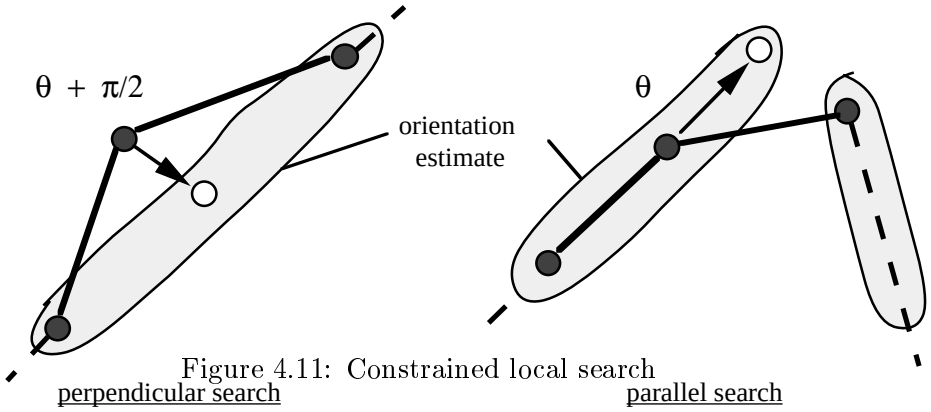


Figure 4.11: Constrained local search

each node given by

$$\boldsymbol{\theta}_{\vec{\chi}} = \sum_{\vec{\psi} \in \mathcal{N}_{\vec{\chi}}} \frac{1}{\mathrm{card}\{\Lambda'_{\vec{\chi},\vec{\psi}}\}} \sum_{m,n \in \Lambda'_{\vec{\chi},\vec{\psi}}} \boldsymbol{\theta}_{mn}(M) \tag{4.48}$$

where $\Lambda'_{\vec{\chi},\vec{\psi}}$ is the set of pixels that lie under the link $\lambda(\vec{\chi},\vec{\psi})$. A search is made first in the perpendicular direction $\theta_{\vec{\chi}} + \frac{\pi}{2}$ and the cost given by equation 4.46 is evaluated at each new position. The cost is anisotropic in the sense that it is dependent on the energy of the local orientation.

If no better position is found then the parallel direction $\theta_{\vec{\chi}}$ is searched. Figure 4.11 shows the effect of the two types of local search on the node positions. The perpendicular movement adjusts the node positions along curves while the parallel movement is aimed to force nodes towards the vertices of corner features.

The idea of using position and orientation to derive a local cost function, which is then minimised, has been used in a relaxation labelling framework by Parent and Zucker [76], or through dynamic programming by Amini *et al* [2]. This process of boundary relaxation forms part of the split-and-merge, region growing and edge detection approach reported by Liow and Pavlidis [66] [77]. All these workers modify the local contour to maximise a 'merit' based on position, orientation, plus a third term based on the curvature, which is

not explicitly present in the cost function of equation (4.46). The curvature term is aimed at improving the smoothness of the boundary, which is achieved in this work by the node elimination step.

2. Constrained Search: Isotropic Cost Component

To help the inference of corner features and improve the distribution of nodes along curved boundary segments, there is a secondary movement of the nodes. The direction of this is determined by the resultant *force* vector between a node and its neighbours. This vector is given by

$$\mathbf{f}_{\vec{\chi}} = \sum_{\vec{\psi} \in \mathcal{N}_{\vec{\chi}}} \frac{\max[\lambda(\vec{\chi}, \vec{\psi})]}{\lambda(\vec{\chi}, \vec{\psi})} \begin{pmatrix} \cos(\arg(\lambda(\vec{\chi}, \vec{\psi})) + \pi) \\ \sin(\arg(\lambda(\vec{\chi}, \vec{\psi})) + \pi) \end{pmatrix} \tag{4.49}$$

where $\arg(\lambda(\vec{\chi}, \vec{\psi}))$ is the angle of the link. The contribution to the total cost from each neighbour is inversely proportional to its distance from $\vec{\chi}$. In other words, the shorter the link, the greater the contribution to the resultant vector which results in a 'repulsive force' being created between nodes. The movement is then effected in the projection of this vector on to the local orientation. There is *no* constraint on the new position being better in cost terms which has the effect of disturbing nodes out of local minima. This movement is kept small, however, and only allowed if no constrained movement, from step 1, has occurred.

Figure 4.12 illustrates 2 cases where the 'repulsive force' takes effect. In case (a) node $\vec{\chi}$ is moved in direction $\mathbf{m}$, which will encourage $\vec{\psi}$ to move closer to the vertex of the corner at the next iteration. In case (b) the movement of $\vec{\chi}$ along $\mathbf{m}$ will even out the distribution of the nodes along the curve segment.

This process is not as intuitive as the constrained search of step 1, but it is essential, however, for corner features to be correctly inferred. As has already been noted, there is no orientation information around the corners and branch points even though they are locally the best positions for the nodes to lie.

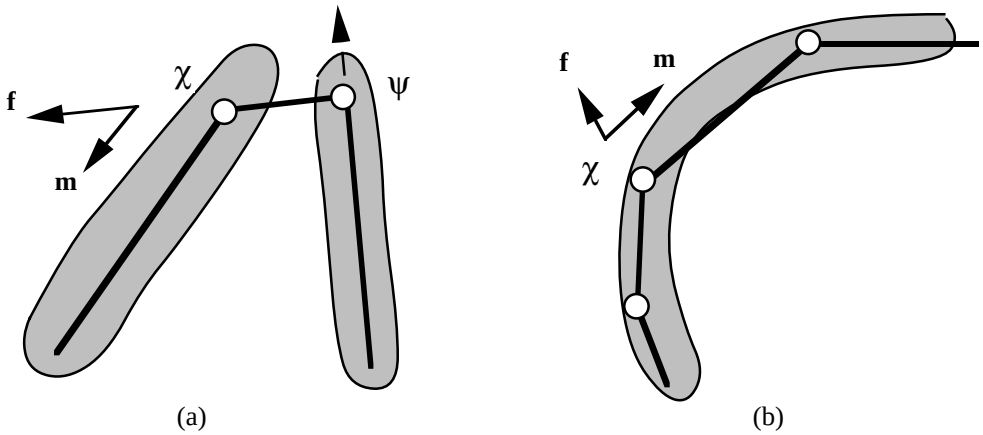


Figure 4.12: 'Repulsive force' movement

3. Node Elimination

The node elimination step aims to develop straight line features by removing nodes that are adding no more information to the segmentation. This step has been incorporated directly into the node processing, rather than being left as a separate post-process, as the increasing link lengths which result invariably improve the position estimates of nodes. This is particularly important for low signal to noise images and those containing straight line features.

The node elimination further helps towards the realisation of the image model where the region boundaries are represented by as few nodes as necessary. It is difficult, however, to determine how a particular boundary representation is optimal in this sense for a given image under varying degrees of noise. For simple polygonal object/background data it is easy to say what a perfect segmentation is, but with more natural scenes without prior knowledge the determination of an optimal representation is more problematic.

A straight line hypothesis is used to determine whether a node should be eliminated. The average orientation vectors along each boundary link of the node under consideration are compared against the hypothesised straight line directly connecting the two neighbours.

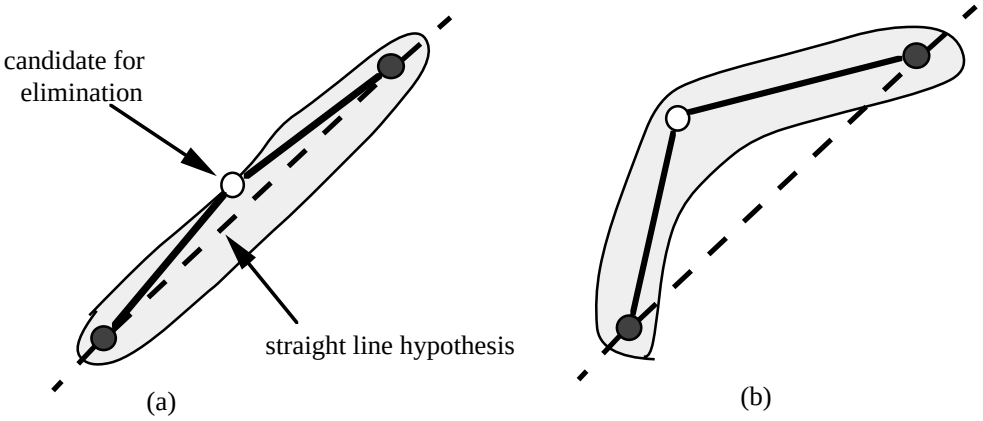


Figure 4.13: Node elimination

The average vectors are decomposed into components along and perpendicular to the hypothesised straight line. The parallel component is a ML estimate of the signal and the orthogonal component due to the noise. The test, therefore, consists of comparing the orthogonal component against some proportion of the estimated noise standard deviation. A node is eliminated if the average, projected component is less than a proportion of the noise s.d.

Figure 4.13 shows the two possible cases: (a) where the hypothesis may be valid, depending on the noise in the data and (b) where the hypothesis is invalid as the node lies on a curve.

The straight line test is thus tuned to the noise in the data. This corresponds well to what is perceived by the HVS without prior knowledge that a particular feature is a straight line. If the SNR is low then almost any boundary shape is possible: it is difficult to determine whether the boundary is really curved or straight. In this case a straight line seems to be the better choice. At high SNRs a straight line hypothesis should be less readily acceptable because more complicated boundaries are visible.

4. Node Addition

The node addition step is the converse process to node elimination and is aimed at improving the boundary representation along high curvature, or rapidly changing features. Nodes are sited at mid-points of links where the boundary segment does not adequately describe the underlying orientation data. A consistency measure is calculated that indicates the variability of the orientation vectors under the link [61] [93]

$$c(\vec{\chi},\vec{\psi}) = \frac{\left|\sum_{mn\in\Lambda'_{\vec{\chi},\vec{\psi}}} \boldsymbol{\theta}_{mn}(M)\right|}{\sum_{mn\in\Lambda'_{\vec{\chi},\vec{\psi}}} |\boldsymbol{\theta}_{mn}(M)|} \tag{4.50}$$

which is the magnitude of the average vector over the average magnitude and the range of c is between 0 and 1 where 0 is inconsistent and 1 is consistent. Nodes are added if this measure is below some threshold ($c < 0.75$ was used in the experiments presented below).

The node addition step is performed after the node elimination so that the newly created nodes have time to move to a lower cost position before the next elimination step. The next node elimination step will remove any additional nodes which have been wrongly added so that an over generous addition threshold is compensated for by the elimination which is noise adaptive.

Summary

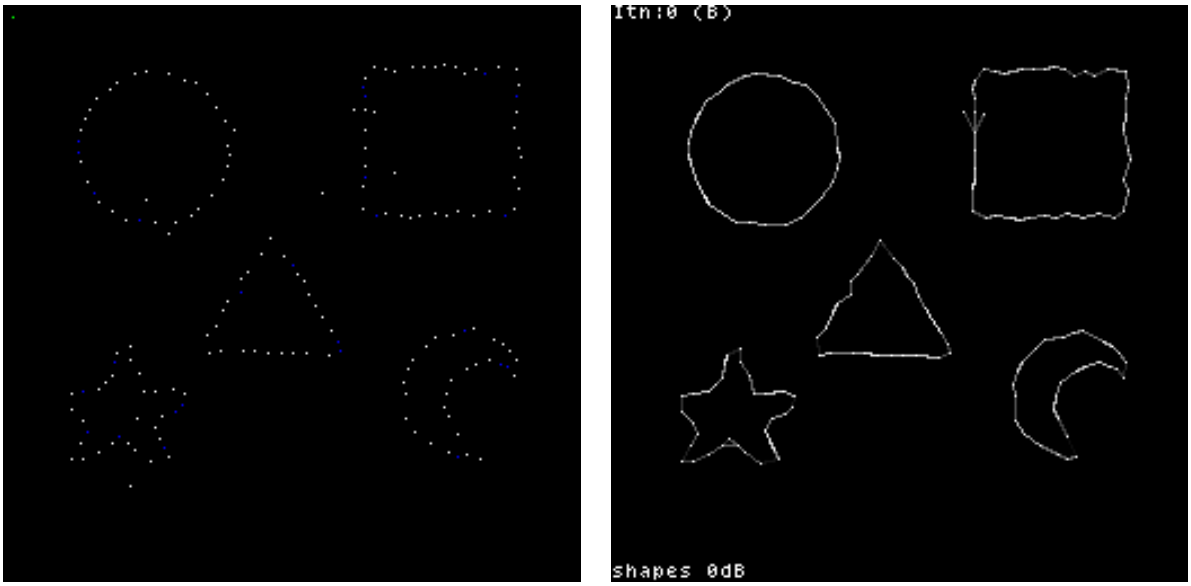
The steps of the boundary refinement process are summarised in figure 4.14, where the input to the procedure is an orientation pyramid.

4.3.4 Results

The results of the node selection step and boundary linking are illustrated in figure 4.15 for the orientation pyramid of the 'shapes' 0dB image (figure 4.4). The boundaries of all the regions have been successfully extracted and joined, with just one or two spurious nodes and links. The subsequent processing is illustrated in figure 4.16 for the bottom left hand

```
procedure Boundary_Refinement (Orient_Pyramid)

begin

    X = Boundary_Node_Selection (Orient_Pyramid);
    B = Boundary_Node_Linking (Orient_Pyramid, X);

    while not Converged (B) do
    begin
        for each node c in B do
        begin
                Moved = false;
                Moved = Constrained_Search (c);
                if not Moved then Repulsive_Force (c);

                Node_Elimination (c);
                Node_Addition (c);
        end

        B' = Update_Positions (B);
        B  = B';
    end

end
```

Figure 4.14: Boundary Refinement Process

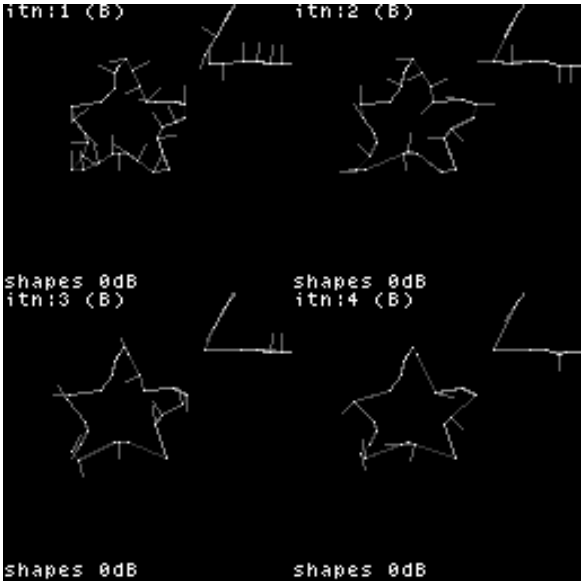


(a) Selected boundary nodes (b) Initial boundary graph

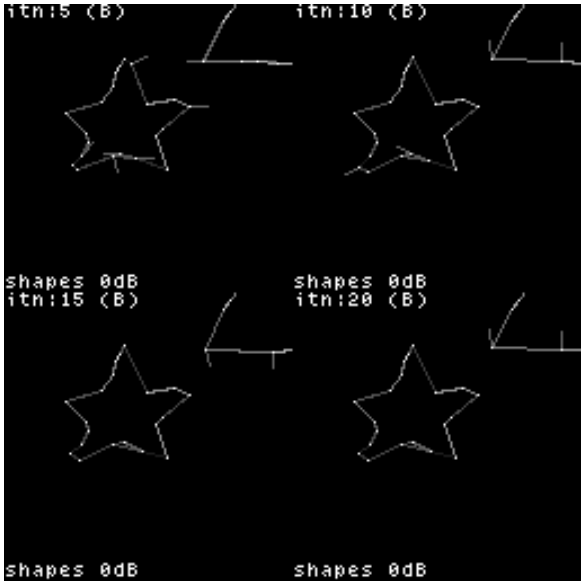
Figure 4.15: Boundary estimation on 'shapes' 0dB

quadrant of the original image. The direction of movement of the nodes at each iteration is shown by vectors. Many of the nodes have moved perpendicular to the local boundary in the early iterations. The effect of the node elimination procedure is also visible by the reduction in the total number of nodes along a given boundary segment. The result, shown in figure 4.16(c), after 20 iterations show that the processing has estimated the boundaries of the square and triangle quite well. The curves are perhaps a bit too coarse, but this is clearly a result of the polyline boundary model. The performance on the corners of the star and crescent is encouraging considering the level of noise present in the original.

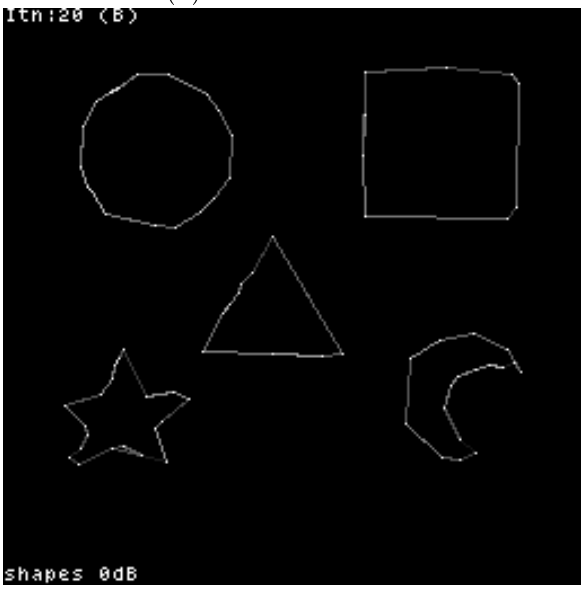
The boundary segmentation was also performed on the 'table 1' image (figure 3.16). Figure 4.17(a) shows the image level of the orientation pyramid. The processing after 1 and 10 iterations is shown to demonstrate the effectiveness of the node elimination reducing many of the straight edges in the image to single curve segments, for example, the vertical edges of the mug on the left of the image.



(a) Iterations 1 to 4

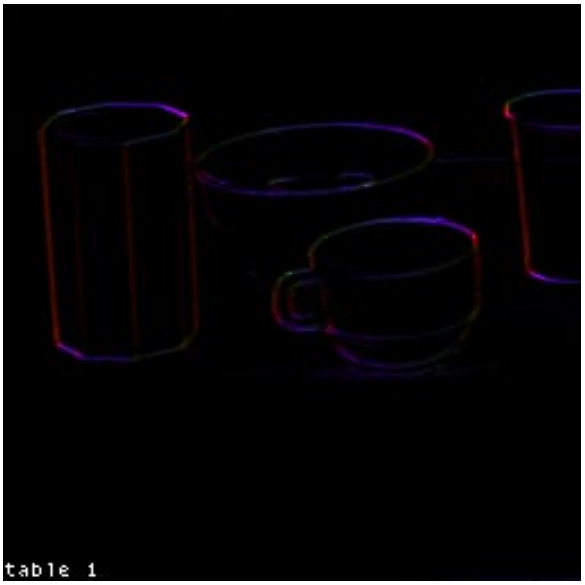


(b) Iterations 5 to 20

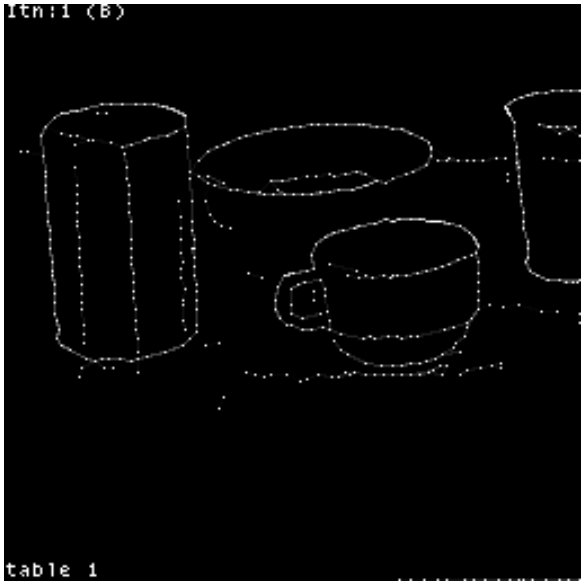


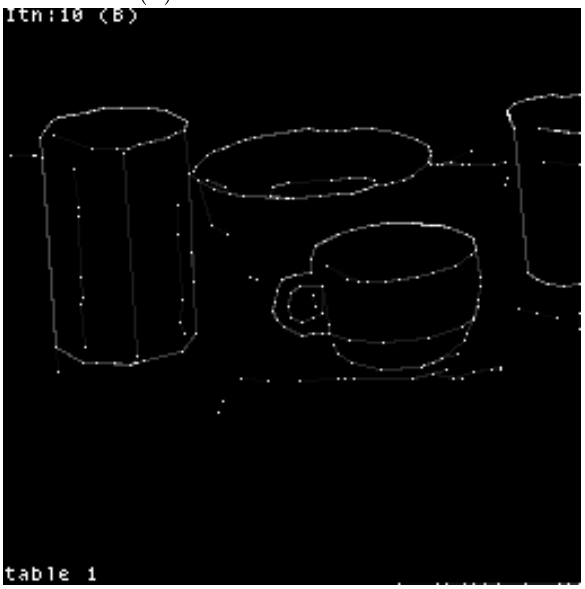
(c) Result after 20 iterations

Figure 4.16: Boundary estimation on 'shapes' 0dB cont.



(a) Orientation estimate





(b) Boundary after 1 iteration

(c) Boundary after 10 iterations

Figure 4.17: Boundary estimation on 'table 1' image

Chapter 5

Process Interaction and Results

5.1 Introduction

This chapter describes the interaction process that combines relevant information from the region and boundary processes within the framework of the model developed in Chapter 2. The interaction process allows the flow of information between the two otherwise autonomous processes and is necessary to overcome the inherent limitations of using either process alone.

The second part of this chapter is devoted to the presentation and discussion of results obtained from the cooperative segmentation method. Results are presented for both synthetic and natural images at various signal to noise ratios. There are comparative results using an implementation of the the Spann and Wilson algorithm [88] [86], and an analysis of the new method is presented using measurements of pixel classification and boundary error as a function of the input signal to noise ratio.

5.2 Process Interaction

From the overall structure of the segmentation method outlined in Chapter 2 (figure 2.5), there are clearly 2 data paths in the interaction process: a flow of information from the boundary to the region process and the reverse flow, from the region to the boundary

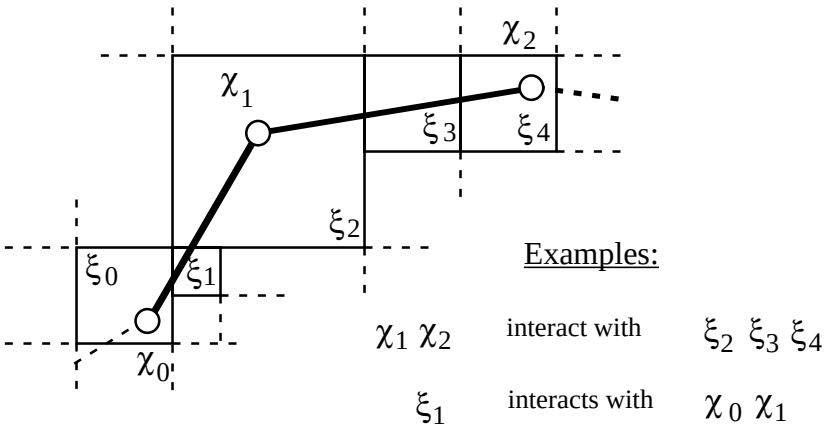


Figure 5.1: Interaction neighbourhood

process. The common denominator between the two processes is the estimate of the orientation and position, so this seems like the logical vehicle for the interaction. Since the two processes can run independently, it is important that the interaction interleaves them without disruption of either. There is thus a synchronisation problem. Also there is need to exchange only reliable information as otherwise there is the danger of a positive-feedback situation where unreliable information continues to be recycled between the two processes.

The interaction takes place in a neighbourhood which is defined in terms of the intersection of the boundary *links* and the region blocks. Figure 5.1 shows a typical interaction neighbourhood. In this example boundary link $(\vec{\chi}_0, \vec{\chi}_1)$ is defined to interact with all the blocks that it passes through, namely $\{\vec{\xi}_0, \vec{\xi}_1, \vec{\xi}_2\}$. Similarly, the region node $\vec{\xi}_2$ is defined to interact with all the boundary links that pass through it, namely $\{(\vec{\chi}_0, \vec{\chi}_1), (\vec{\chi}_1, \vec{\chi}_2)\}$ and hence the boundary nodes $\{\vec{\chi}_0, \vec{\chi}_1, \vec{\chi}_2\}$. In the following sections the interaction neighbourhoods are denoted by $\mathcal{I}$.

5.2.1 Boundary-to-Region Interaction

In the boundary to region interaction, the region edge estimates are combined with the local boundary orientation and position. The best linear combination is given by the sum

$$\begin{aligned} \boldsymbol{\theta}'_{\vec{\xi}} &= \beta_r \boldsymbol{\theta}_{\vec{\xi}} + \beta_b \boldsymbol{\theta}_{\mathcal{I}_{\vec{\xi}}} \\ \mathbf{r}'_{\vec{\xi}} &= \beta_r \mathbf{r}_{\vec{\xi}} + \beta_b \mathbf{r}_{\mathcal{I}_{\vec{\xi}}} \end{aligned} \tag{5.1}$$

where β_r and β_b are the combination coefficients and $\mathcal{I}_{\vec{\xi}}$ is the interaction neighbourhood of $\vec{\xi}$ so that $\theta_{\mathcal{I}_{\vec{\xi}}}$ represents the orientation of the local boundary.

Combining Orientation Estimates

The combination coefficients can be derived by considering the best combined estimate of two data given their respective variances. Given the data x_r and x_b such that

$$\begin{aligned} x_r &= \alpha_r s + n_r \\ x_b &= \alpha_b s + n_b \end{aligned} \tag{5.2}$$

the best estimate

$$\hat{s} = \beta_r x_r + \beta_b x_b \tag{5.3}$$

of the normalised variable s is given by the orthogonality principle. From

$$\begin{aligned} E[s x_r] &= \beta_r E[x_r^2] + \beta_b E[x_r x_b] \\ E[s^2] = 1 \Rightarrow \alpha_r &= \alpha_r^2 \beta_r + \sigma_r^2 \beta_r + \alpha_r \alpha_b \beta_b \end{aligned} \tag{5.4}$$

and from

$$\begin{aligned} E[s x_b] &= \beta_r E[x_r x_b] + \beta_b E[x_b^2] \\ \Rightarrow \alpha_b &= \alpha_r \alpha_b \beta_r + \alpha_b^2 \beta_b + \sigma_b^2 \beta_b \\ \Rightarrow \beta_b &= \frac{\alpha_b - \alpha_r \alpha_b \beta_r}{\alpha_b^2 + \sigma_b^2} \end{aligned} \tag{5.5}$$

where

$$\begin{aligned} \sigma_r^2 &= \mathrm{var} n_r \\ \sigma_b^2 &= \mathrm{var} n_b \end{aligned} \tag{5.6}$$

Substituting for β_b from equation (5.5) into equation (5.4) yields

$$\begin{aligned} \beta_r &= \frac{\alpha_a \sigma_b^2}{\alpha_r^2 \sigma_b^2 + \alpha_b^2 \sigma_r^2 + \sigma_r^2 \sigma_b^2} \\ \beta_b &= \frac{\alpha_b \sigma_r^2}{\alpha_r^2 \sigma_b^2 + \alpha_b^2 \sigma_r^2 + \sigma_r^2 \sigma_b^2} \end{aligned} \tag{5.7}$$

However the estimate $\hat{s}$ is biased since

$$\begin{aligned} E[\hat{s}] &= E[\beta_r x_r + \beta_b x_b] \\ &= (\beta_r \alpha_r + \beta_b \alpha_b) E[s] \end{aligned} \tag{5.8}$$

By making

$$\beta_r \alpha_r + \beta_b \alpha_b = 1 \tag{5.9}$$

an unbiased estimate is given by the coefficients

$$\begin{aligned} \beta_r' &= \frac{\beta_r \alpha_r}{\beta_r \alpha_r + \beta_b \alpha_b} \\ \beta_b' &= 1 - \beta_r' \end{aligned} \tag{5.10}$$

The variance of the estimate is given by

$$\begin{aligned} \mathrm{var}\hat{s} &= E[(s - \hat{s})^2] \\ &= E[s^2] - 2E[s\hat{s}] + E[\hat{s}^2] \\ &= 1 - 2E[s(\beta_r x_r + \beta_b x_b)] + E[(\beta_r x_r + \beta_b x_b)^2] \end{aligned} \tag{5.11}$$

Using equations (5.2), (5.9) and (5.6) this reduces to

$$\mathrm{var}\hat{s} = \beta_r^2 \alpha_r^2 \sigma_r^2 + \beta_b^2 \alpha_b^2 \sigma_b^2 \tag{5.12}$$

The amplitude coefficients α_r and α_b are obtained from estimates of the variances of the data

$$\begin{aligned} \mathrm{var}x_r &= \alpha_r^2 + \sigma_r^2 \\ \Rightarrow \alpha_r^2 &= \mathrm{var}x_r - \sigma_r^2 \end{aligned} \tag{5.13}$$

and similarly

$$\alpha_b^2 = \mathrm{var}x_b - \sigma_b^2 \tag{5.14}$$

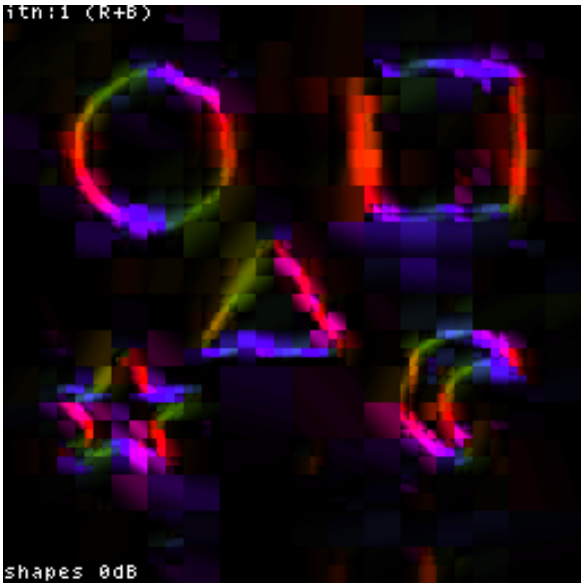
Estimates are combined after the neighbourhood averaging of the region based edge estimates of section 3.5.1. The boundary information thus influences the calculation of the prior region link probabilities (equation (3.20)) and therefore the link decision process of equation (3.13).

Figures 5.2(a) to (d) show the effect of combining the two estimates of orientation. Each result shows the region based orientation estimates which have been *synthesised* in each region block after combining with the local boundary over several iterations. It is clear from these figures that the boundary process greatly improves the estimates by both localising the edges better and reducing the noise in the region interiors.

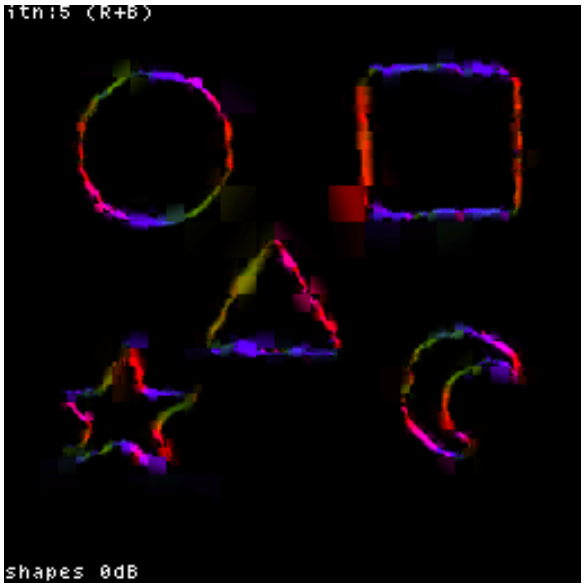
5.2.2 Region-to-Boundary Interaction

The reverse data flow is done through the addition of an extra term to the boundary cost function of equation (4.46) which introduces region information into the iterative refinement. The modified cost function takes the following form

$$\begin{aligned} C'(\lambda(\vec{\chi},\vec{\psi})) = \frac{1}{\mathrm{card}\{\Lambda'_{\vec{\chi},\vec{\psi}}\}} \Bigg[& \beta''_b \sum_{m,n\in\Lambda'_{\vec{\chi},\vec{\psi}}} |\boldsymbol{\theta}^B_{mn}(M)| \cos\Big(0.5\,\arg(\boldsymbol{\theta}^B_{mn}(M)) - \arg(\lambda(\vec{\chi},\vec{\psi}))\Big) \\ & + \beta''_r \sum_{m,n\in\Lambda'_{\vec{\chi},\vec{\psi}}} |\boldsymbol{\theta}^R_{mn}(M)| \cos\Big(0.5\,\arg(\boldsymbol{\theta}^R_{mn}(M)) - \arg(\lambda(\vec{\chi},\vec{\psi}))\Big) \Bigg] \end{aligned} \tag{5.15}$$

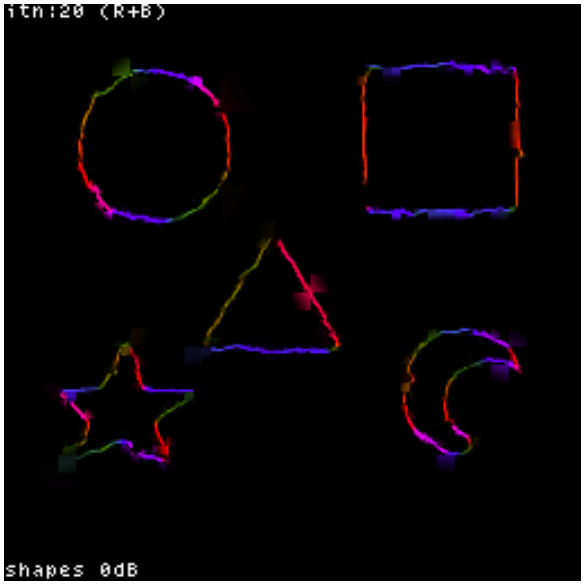


(a) Iteration 1



(b) Iteration 5

(a) Iteration 10



(b) Iteration 20

Figure 5.2: Combined region and boundary orientation estimates for 'shapes' 0dB image, iterations 1, 5, 10, 20

where the orientation estimates have been superscripted by B and R to indicate their source i.e. either boundary or region based. This modified cost function is thus a linear combination of the two orientation energies, where the relative weighting factors β_b'' and β_r'' are obtained in the same manner as above. Thus the influence that either estimate has on the positions of the nodes is controlled by the respective estimates of the noise variances.

5.2.3 Corner Blocks

The region edge estimates cannot be relied upon in the vicinity of corner features because of the simplicity of the edge model, therefore it is desirable to deal with such blocks using information from the boundary process, which is better able to infer corner features. Region blocks which contain corner features must also be dealt with differently during the block splitting (section 3.5.2), which again is based on the assumption of a straight edge through the block.

It is relatively easy to identify whether a particular region block should be considered as a corner block by looking at the boundary nodes that lie in the block. If these nodes lie at point of high curvature then they are treated as vertex nodes and the region block is marked as being a corner block. The 'high curvature' test used in the experiments is based simply on the acuteness of the angle made by the boundary links at the node with a threshold angle of $2\pi/3$. Having thus identified corner blocks, the region block splitting is modified so that the origin of the shrinking process, that provides the estimates of the children, is set to be the position of the corner boundary node. Figure 5.3 illustrates how this modification provides the estimates from the children still using the edge 'focusing' technique described in section 3.5.2.

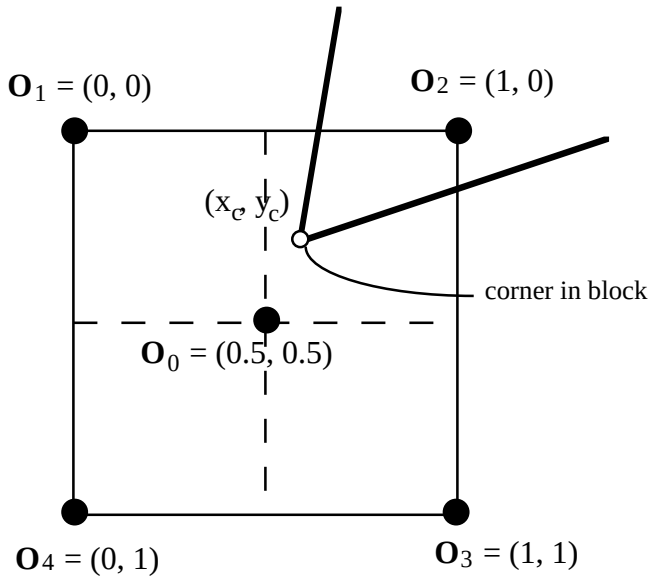


Figure 5.3: Spatial shrinking for corner blocks

5.2.4 Node Insertion

Where there is evidence of a boundary from the region process and no boundary present in the interaction neighbourhood, as defined above, boundary nodes are inserted into the boundary graph. The criterion for creating new boundary nodes is that a region block has been split and there are no boundary nodes in the block. Since the splitting is dependent on a high probability of an edge, $P(\text{edge})$, this provides the evidence for the presence of an edge not found by the boundary process. The created node is then positioned at the mid point of the edge in the region block. It is then linked to nearby boundary nodes, which have degree 1, or other nodes that have also been created, using the local linking method of section 4.3.2. Figure 5.4 shows this insertion process in operation.

The node insertion step has the potential to fill in gaps in the initial boundary process, which is important in low signal to noise ratios where the initial estimate is more uncertain. The dynamic nature of the resulting boundary process also allows the noise thresholds for the boundary node selection (section 4.3.1) to err on the side of caution.

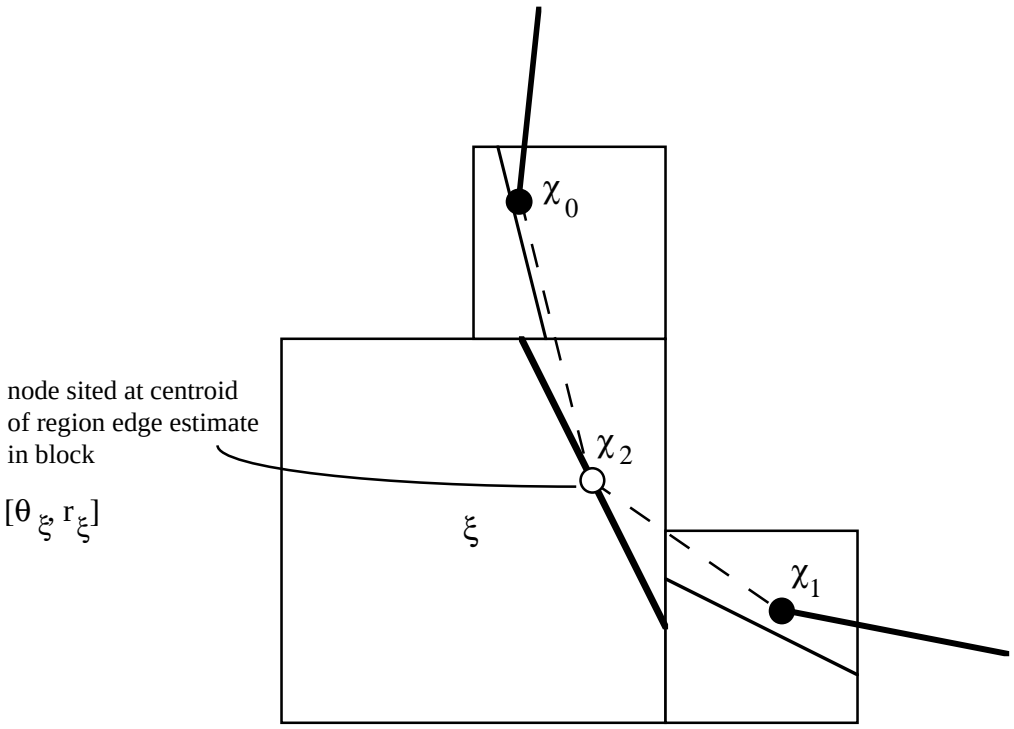


Figure 5.4: Boundary node insertion using region information

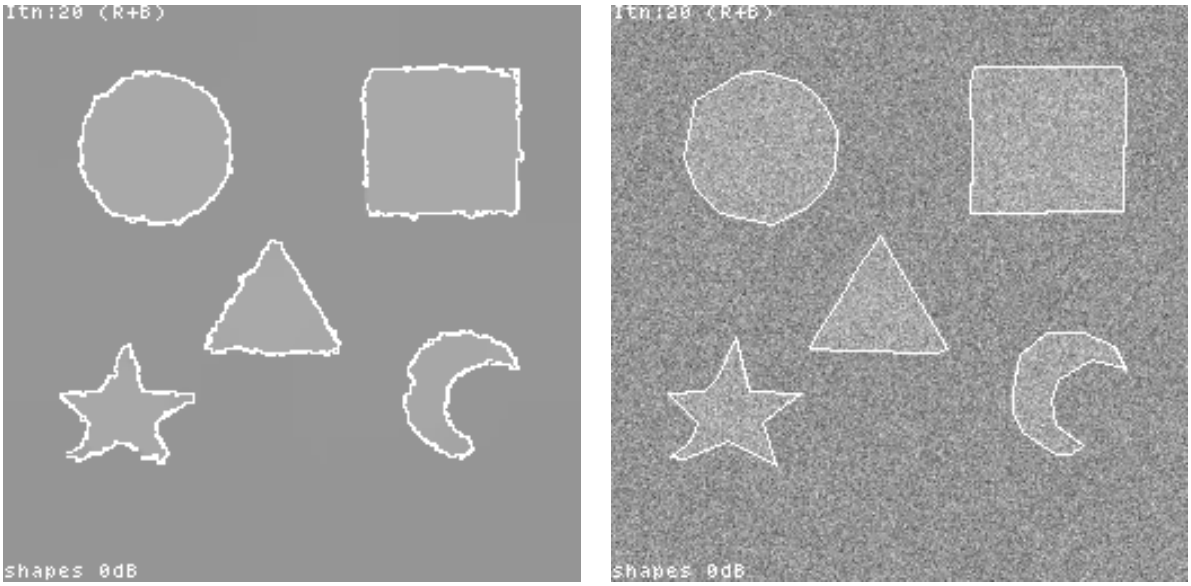
5.3 Results

This section presents the results of the combined segmentation algorithm on a variety of synthetic and natural images. The accuracy of the method is assessed by measuring the pixel classification and boundary errors for a variety of input SNR images. This is compared with results obtained by using the method described in [88].

5.3.1 Synthetic Images

The first result is that for the 'shapes' 0dB image. The region and boundary segmentation is shown separately in figure 5.5 and can be compared with the separate processing results in Chapter 3 and Chapter 4. In figure 5.5(b) the boundary result has been overlayed on to the original and shows a remarkable performance on all the shapes.

To assess the effect of varying the degree of noise, differents amounts of noise were added



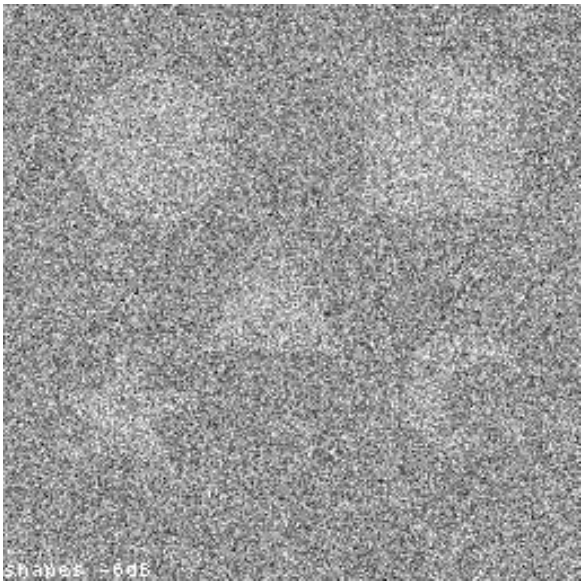
(a) Region segmentation (b) Boundary segmentation

Figure 5.5: Combined segmentation on 'shapes' 0dB

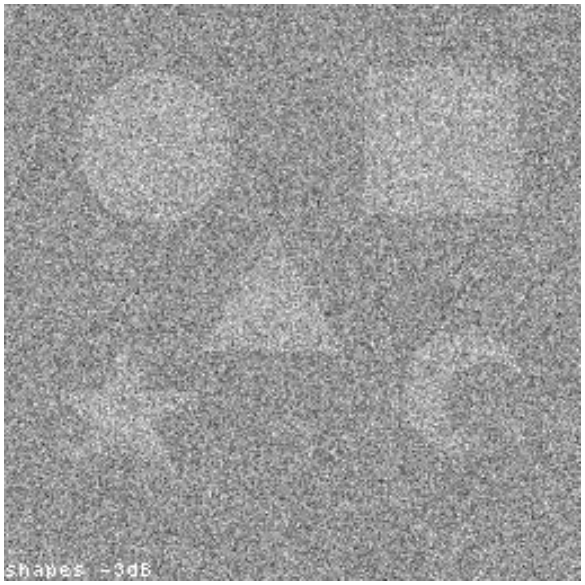
to the 'shapes' image. These input images are shown in figure 5.6. The -6dB level of noise makes it hard to discern the boundaries of the star and crescent, even knowing what to look for, whereas, not surprisingly, the more 'compact' circle and square are less affected. Figure 5.7 shows the results on these images.

The results on the -6dB input image (figure 5.7(a)) has only just managed to capture the rough outlines of the triangle, star and crescent. There are also many spurious boundary segments in the background. Nevertheless, the circle and square have been reasonably segmented, and a few of the more acute corner features also detected. At -3dB (figure 5.7(b)) there is more success on all the shapes. The results on the two higher SNR images (3dB and 6dB) are almost perfect.

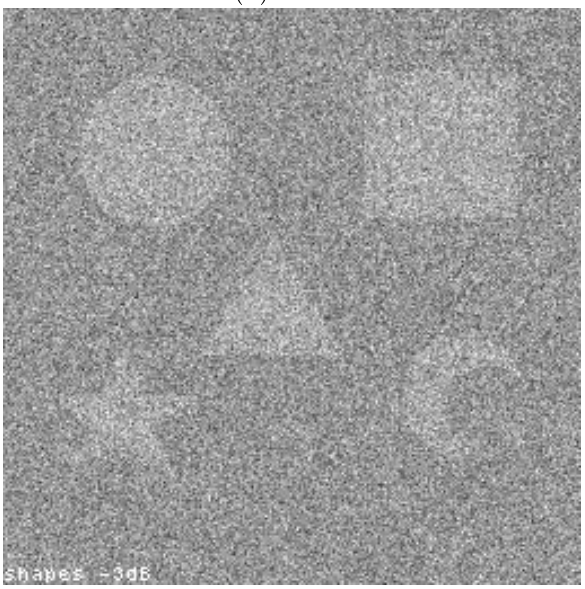
Figure 5.8(b) shows the results on the 'blobs' image at 0dB. This image has regions which are of varying sizes and some have been deliberately located close together to test the separation abilities of the segmentation. The result shows that the segmentation method can cope well with the varying size, but not so well with the regions being close together.



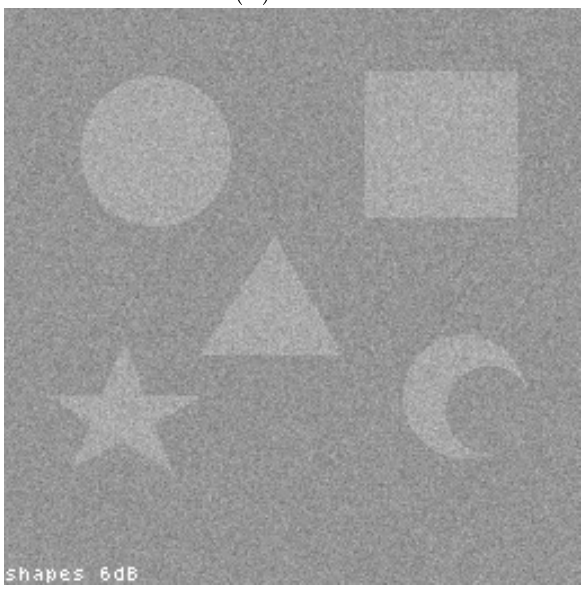
(a) -6 dB



(b) -3 dB

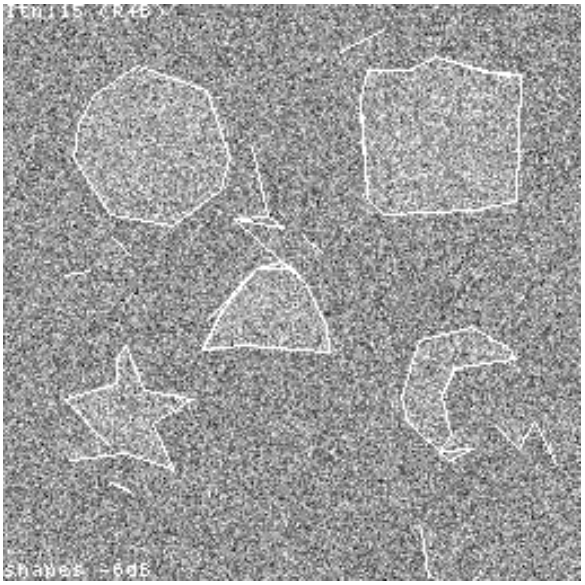


(c) 3 dB

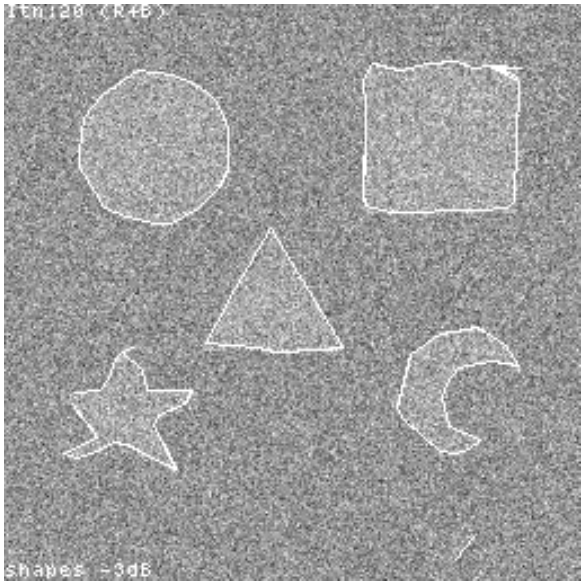


(d) 6 dB

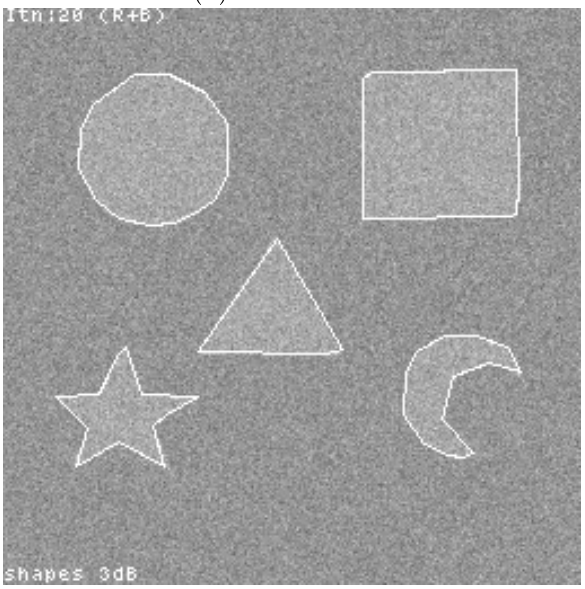
Figure 5.6: 'shapes' at a variety of SNRs



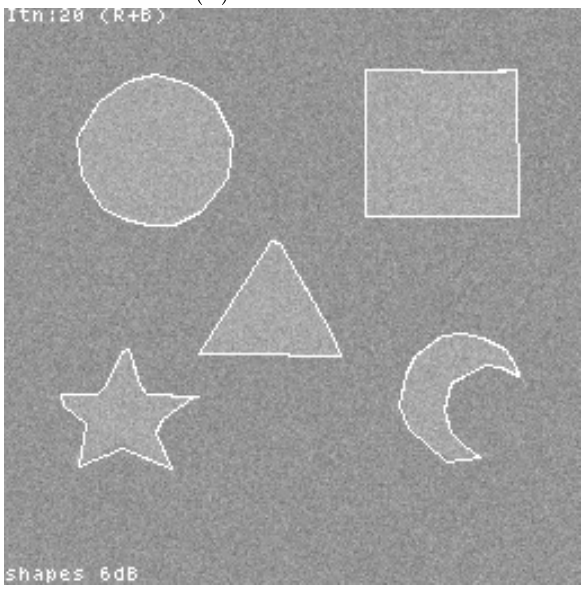
(a) -6 dB result



(b) -3 dB result



(c) 3 dB result



(d) 6 dB result

Figure 5.7: Combined segmentation on 'shapes' at a variety of SNRs

The boundary has been linked between some circles because it has been fooled by the local orientation information.

To test out the method on more random boundaries, it was applied to the 'widgets' image (figure 4.5). The results of the region and boundary segmentations are shown in figures 5.8(c) and (d). Although the region segmentation has estimated the bottom right shape to be separate from the background, the boundary process linking, and any node insertion step, has failed to close the boundary.

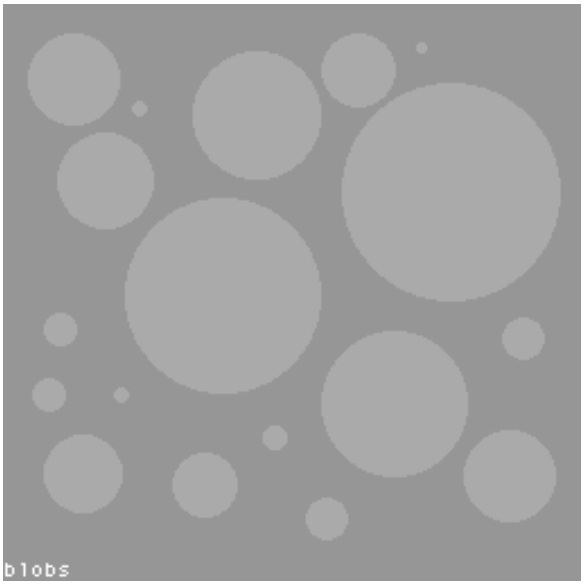
5.3.2 Comparative Results

The combined segmentation algorithm was assessed by using two measures of performance [17]:

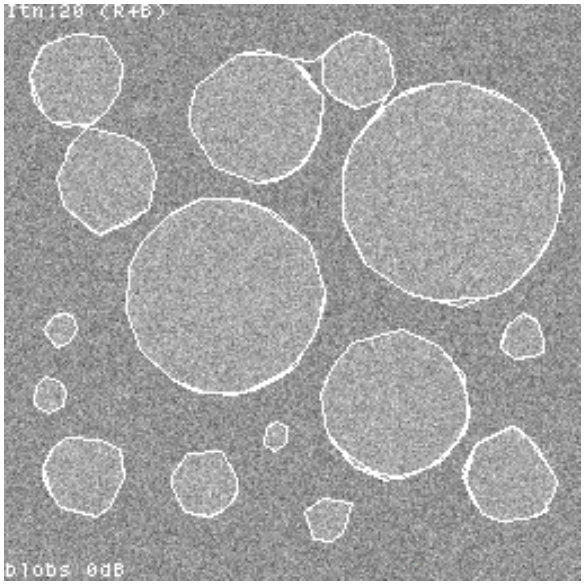
1. *Pixel classification error* which is derived by counting all the misclassified pixels and is expressed as a percentage of the total number of image pixels ($256 \times 256 = 65536$).

2. *Root mean square (RMS) boundary error* which is obtained by averaging the squared boundary error, i.e. the square of the distance (in pixels) between the estimated boundary and the true boundary, and taking the square root of this value. This error emphasises large boundary differences, and therefore lies between the mean and the maximum error.

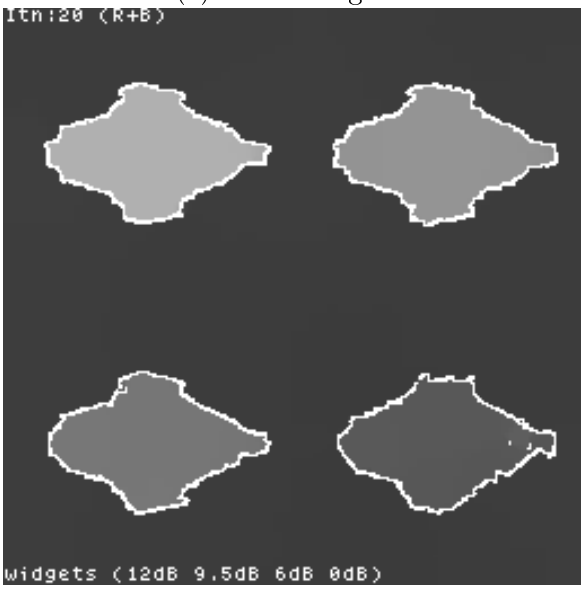
For simple bipartite images, where the boundary varies along one or other axes, the calculation of the boundary error is straightforward. With images containing complex region shapes, such as the 'shapes' test image used in this work, a more complicated approach is needed. The correct boundary error is obtained by measuring the difference between the estimated and true boundary *orthogonal* to the true boundary. An orientation estimation from the noise free test image was used to guide the boundary measure. Figure 5.9(a) shows the boundary error being measured on the 'shapes' -6dB result by the Spann and Wilson method. The estimated boundary is shown in white and the nearest true boundary



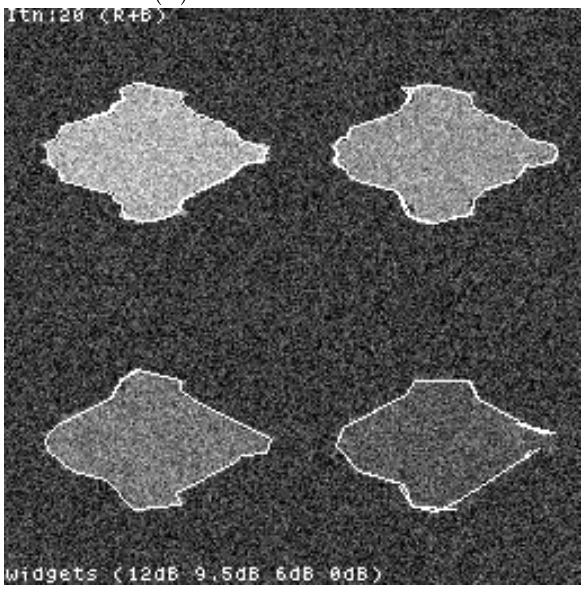
(a) 'blobs' original



(b) 'blobs' 0 dB result

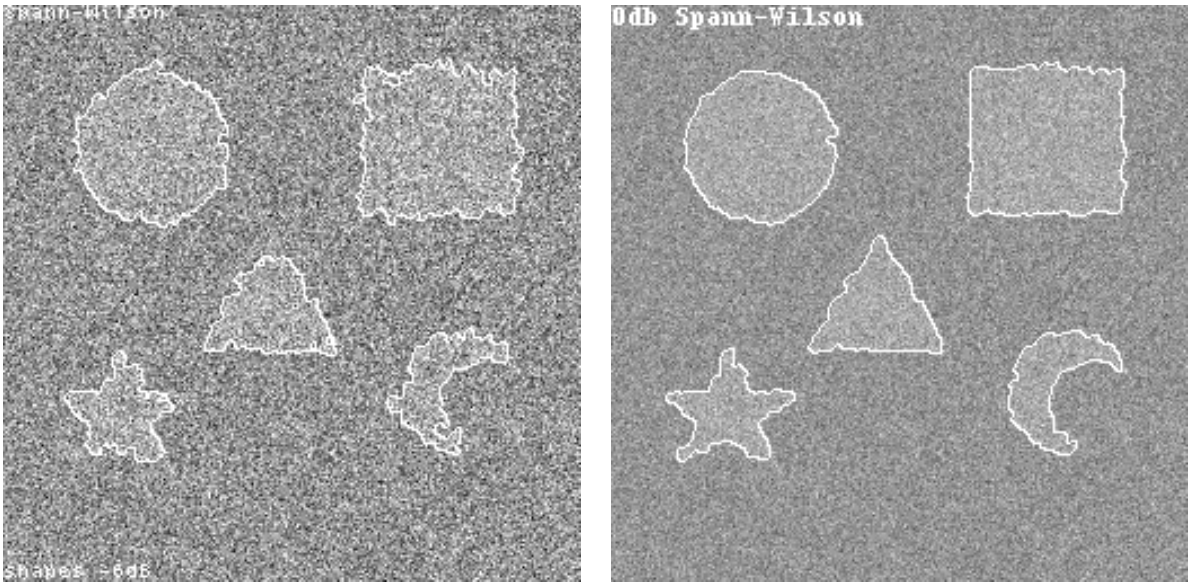


(c) 'widgets' region result



(d) 'widgets' boundary result

Figure 5.8: Combined segmentation on 'blobs' 0dB and 'widgets' 12dB-0dB



(a) 'shapes' -6 dB (b) 'shapes' 0 dB

Figure 5.9: Comparative results using Spann and Wilson method

is marked in black.

The two measures of performance were run on a set of results on the 'shapes' test image at inter-regional SNRs of -6dB to 12dB, at 3dB intervals. An implementation of the Spann and Wilson segmentation method [88] [86] was used for comparison. Figure 5.9(b) shows the comparative result for the 'shapes' 0dB image. Graphs of the pixel classification and boundary errors are shown in figures 5.10 and 5.11, where the dashed line is the results of the Spann-Wilson method. Figure 5.10 shows the new method to be comparable, in terms of pixel classification error, with the Spann and Wilson algorithm. In terms of RMS boundary error, however, the results shown in figure 5.11 confirm the qualitative findings that the new method is superior at locating boundaries. Both the measures show the new method to rapidly degrade in performance at SNRs below 0dB.

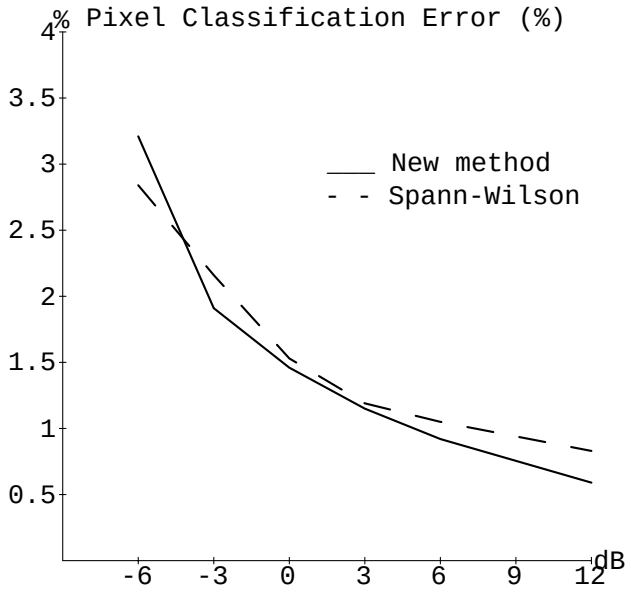


Figure 5.10: Percentage pixel classification error on 'shapes'

5.3.3 Natural Images

The figures in this section (figure 5.12 to figure 5.14) show the results of running the combined segmentation on 3 natural images.

Figure 5.12 shows the combined segmentation result for the 'table 1' image (figure 3.16). These can be compared with the region and boundary only processing of Chapter 3 and Chapter 4.

The 'table 2' image (figure 5.13(a)) is similar to the 'table 1' image and was chosen as it also has several flat regions and shapes with polygonal sides. Satisfactory results for these images are obtained. In figure 5.13(c) all the major curves are present. The places where there are missing curve segments, such as where the large bowl on the left meets the juice carton, corresponds to the weak orientation information in this area (figure 5.13(b)). The

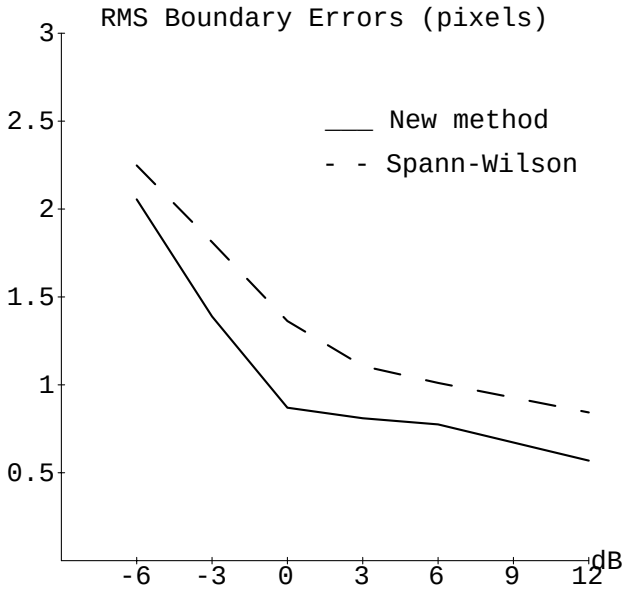
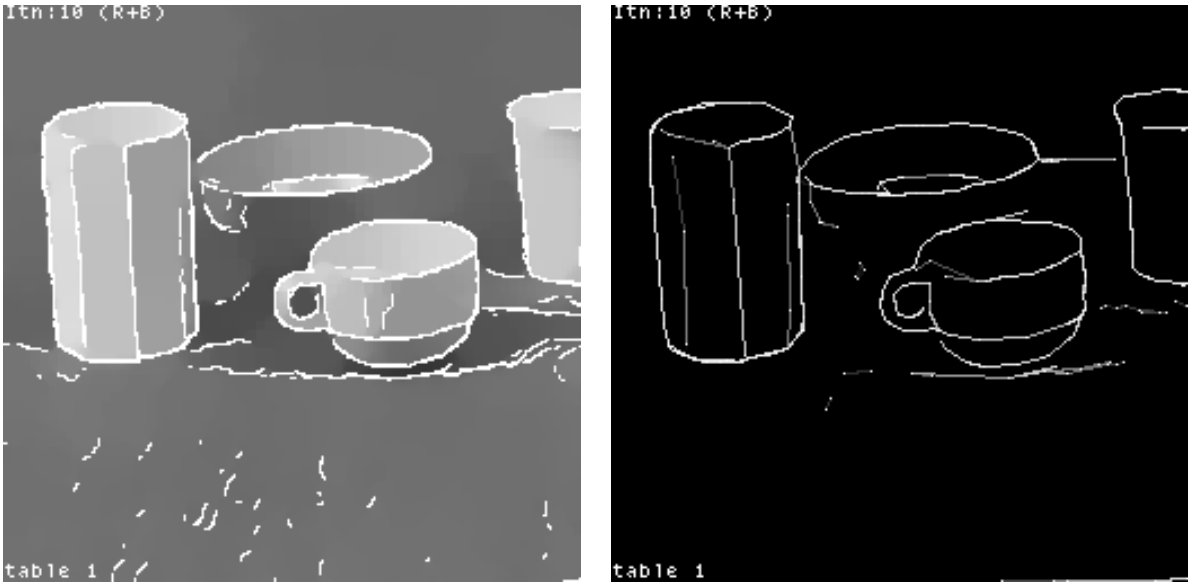


Figure 5.11: RMS boundary error on 'shapes'

edges of the shadows on the table also cause some problems.

The 'girl' image (figure 1.1(b)) contains many more problems than the table images. The results on this image are shown in figure 5.14. The majority of the significant curves have been detected. It is more difficult to interpret the result of the region segmentation other than to say that the main image regions are represented by a homogeneous colour. There are clearly problems in the textured regions, such as the feathers that hang off the hat and the hair. The curves of the mirror, hat and the vertical straight edges in the background have been accurately found. However, less success has been achieved with the facial features partly because of the high curvature around the eyes and the shallowness of the edge across the nose.

In summary, the results of the cooperative segmentation algorithm compare favourably

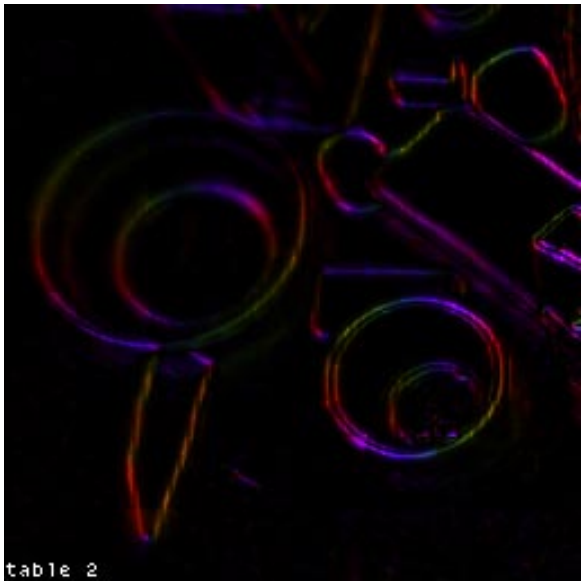


(a) 'table 1' region result (b) 'table 1' boundary result

Figure 5.12: Combined segmentation on 'table 1'

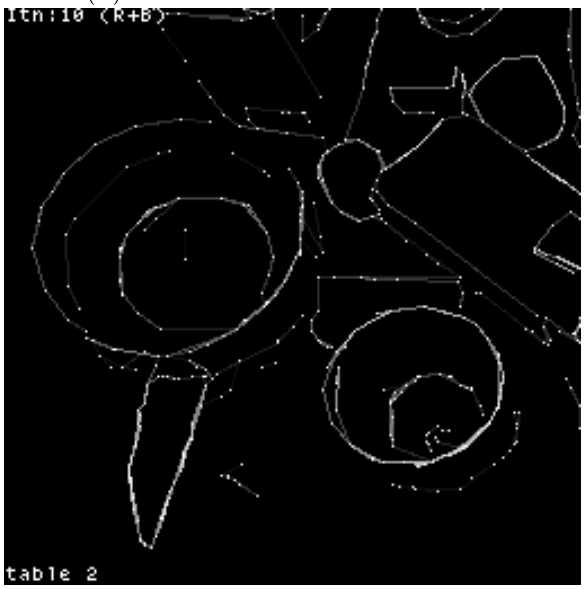
with those reported by Spann [86], on both synthetic and natural images. The quantitative analysis shows the method to be better, in terms of RMS boundary error, over a range of SNRs, at locating object boundaries. This confirms the qualitative findings that the method is superior at capturing complex region shapes even at low signal-to-noise ratios.

(a) 'table 2' original



(b) 'table 2' orientation estimate

(c) 'table 2' region result



(d) 'table 2' boundary result

Figure 5.13: Combined segmentation on 'table 2'

(a) 'girl' orientation estimate

(b) 'girl' region result

(c) 'girl' boundary result

Figure 5.14: Combined segmentation on 'girl'

Chapter 6

Conclusions and Further Work

6.1 Thesis Summary

The work described in this thesis has been concerned with image segmentation within a multiresolution framework. A new algorithm for image segmentation, having a number of interesting features, has been described and shown to be effective in locating regions and their boundaries with high accuracy from noisy data.

In Chapter 1 the fundamental problem of image segmentation was outlined and the importance of the HVS as a model for work in computer vision noted, because of its efficiency and general purpose nature. The task of segmentation was defined as determining 'what is where'. The role of uncertainty in segmentation was discussed; that localisation in class space ('what') conflicts directly with the simultaneous localisation in position space ('where') [109]. It was shown that multiresolution analysis, such as by using image pyramids, allows uncertainty to be tackled through a class/position trade-off.

A review of notable segmentation methods was presented in order to identify common themes in the various approaches and to set a context for this work. The methods were reviewed in three classes: region based, boundary based and combined or cooperative methods, with attention being paid to those that use multiresolution structures.

Region based methods have approached segmentation from a viewpoint of localisation

in class space, with the various methods striving to include a degree of context, based on geometric proximity. Such approaches have often used relaxation techniques and MRF models [8] [10] [111], to estimate region properties e.g relaxation labelling or MAP estimation by Simulated Annealing or Iterated Conditional Modes [54] [9] [39]. It was noted, however, that such methods are supervised, relying on training data for parameter estimation and empirical or *ad-hoc* means to determine prior distributions. MAP estimation in a multiresolution framework, so called Multigrid methods [91] [40], have been successful in overcoming the prohibitive computational burdens involved. The use of MRF models over a range of scales has also been shown to improve long range characteristics of the model.

Boundary based methods have approached segmentation from a viewpoint of positional localisation. A wide variety of methods have been suggested for edge detection, ranging from small kernel gradient operators [41] to multiresolution, frequency domain methods, such as the MFT [20]. It was noted that these methods are also limited by a form of uncertainty which is a trade-off between spatial localisation and noise immunity [22] [74]. Once again success has been achieved by the use of multiple scales.

Combined methods have implicitly or explicitly attempted to integrate region and boundary information, to overcome the inherent limitations of either approach alone. The review highlighted the use of 'line' processes in MRF approaches [38], to model nonstationarities in the image, such as edges. This exploits the natural duality of regions and their boundaries. It was noted, however, that such modifications to the MRF models compound the problems of model identification. Also the line process models are unnecessarily tied to the region model which conflicts with the fact that regions and boundaries, although complementary, nevertheless vary in scale across images. More recent work in cooperative region-boundary methods have suggested a need for a more explicit interaction strategy [77] [66].

A set of requirements for a segmentation approach were proposed in the conclusion to Chapter 1. These were:

1. Prior Knowledge: The problem of parameter estimation and to what degree and in what form higher level knowledge should be incorporated into a segmentation process
2. Local Processing: The processing should reflect the intrinsic structure of images which exhibit features that are local and that their sizes vary across the image.
3. Uncertainty: Both class/position trade-off and localisation/noise-immunity trade-off.
4. Computational Efficiency: The method should require minimum computation.
5. Flexibility of Image Model: The image model should be readily extensible.

In Chapter 2 a framework for image segmentation aimed at fulfilling the above requirements was sought. This chapter considered a model for segmentation derived from a general class of linear multiresolution image models [25] [103], which have been successfully used in image estimation, coding and curve extraction [26] [20] [93]. The main feature of multiresolution models is that it is possible to describe image properties that span a range of scale, which reflects the nature of most images. Small features and fine detail are modelled at relatively high spatial resolutions, while large features are modelled at low spatial resolutions. Such models also have the attraction that they can result in estimation procedures which are computationally efficient.

An adaptation of the generalised multiresolution image model was given which models both region and boundary features. Image generation in the model is defined as a two stage process: first there is a boundary process which uses a vertex based representation to generate a set of connected curves, and second the regions are modelled as an MRF, where the adjacency of the pixels is constrained by the boundaries. At both stages the generation

is a coarse to fine procedure, with evolution of the features being modelled through scale by using conditional probabilities.

A cooperative segmentation algorithm consisting of four main stages was outlined at the end of Chapter 2. The first stage is construction of a pyramid representation of the image, which is considered in more detail in Chapter 3. This forms the input to two separate processes that operate on the region and boundary information respectively. The interaction between the processes, which achieves the segmentation, was defined as the fourth, explicit process.

The region and boundary processes have a common structure which involves scale selection, creation of an adjacency graph and iterative estimation. The scale selection, which is achieved by a process of multiresolution inference, provides an initial segmentation from which an adjacency graph is created. The iterative processing is used to update the initial segmentation using a combination of a MAP decision approach and ML estimation.

Chapter 3 detailed the operation of the region estimation process. The construction and properties of low-pass pyramids were discussed. Local parameter estimation within a pyramid structure was described and the use of a variance estimator to achieve a tessellation of the image shown. The resulting image regions are then linked together to form a region adjacency graph, which is successively refined to achieve a segmentation.

The iterative estimation strategy is based on a Gaussian MRF signal model in which regions are assumed to have independently selected means. The grey level at each selected node is iteratively averaged with its neighbours, if they are believed to belong to the same region. The membership test is a MAP decision that a region link is 'on', given the data at the two nodes it connects. The 'prior' of the link decision process is calculated from estimates of orientation information, which introduces dependence on neighbouring data. Estimates of local orientation are obtained either from region data, by least squares fitting

to a simple edge model, or directly from the boundary processing, described in Chapter 4.

As the estimation proceeds, the initial block segmentation is refined by splitting blocks which are thought to contain edges, and 'focusing' the region information on to the next, higher spatial resolution. The block splitting criterion is based on the probability of an edge in the block and how well the data fits the 2-region hypothesis. A form of fractal contraction transformation is used to focus the region information around the edge to regain spatial localisation.

Chapter 3 was concluded with a description of the convergence characteristics and computational requirements of the region estimation. It was shown that approximately 564 multiplications per pixel are required for 20 iterations of the process. In practice convergence is rapid, typically within 10 iterations. Some experimental results were presented to highlight the operation of the iterative estimation scheme. These illustrated the result of scale selection, the adjacency graph structure and changes of the link states over several iterations.

The boundary estimation process was detailed in Chapter 4. This chapter consisted of two main sections: the first part describing a multiresolution method for estimating orientation, and the second describing the boundary segmentation process.

The importance of orientation as a parameter of an edge was discussed and the benefits of using a double angle representation were outlined. Next a robust method of orientation estimation was described. There are three main steps to the procedure: construction of a grey level pyramid, orientation estimation on each level, and smoothing of the initial estimates by scale-space recursion in the pyramid. The basis of the method is that downward propagation of the more reliable estimates from higher levels of the pyramid to lower levels reduces the estimation errors. It is shown that lateral processing of the propagated estimates, by anisotropic filtering, can further reduce the errors. The results presented

demonstrated the scheme to give significant gains in SNR at little computational cost.

The boundary segmentation process is designed to operate independently of the region estimation process of Chapter 3 and uses an orientation pyramid representation of the image. Like the region process, there are three steps of scale-selection, creation of a boundary (adjacency) graph, and iterative estimation. The node selection process operates top-down on the orientation pyramid of the image. The selected nodes are then linked together using simple, local linking criteria based on orientation, proximity and curvature. The iterative estimation is a relaxation process that allows the selected nodes to be moved to locally 'best' positions. It uses the orientation estimate as an energy landscape from which the best positions are sought by constrained search. Node elimination and addition steps are defined which aim to improve the extracted straight lines and curved boundary segments. Some experimental results were presented that illustrated the operation of the scheme on both synthetic and natural images.

Chapter 5 brought together the region and boundary processes by defining an explicit interaction strategy. It was noted that the logical vehicle for information transfer between the processes was the orientation estimate, i.e. that of the model based region edge estimates and the boundary segment positions. Two data paths were identified: a region-to-boundary path and the reverse, boundary-to-region path. Interaction neighbourhoods were defined between the region and boundary adjacency graphs which exploited the 'duality' between the two representations.

By considering the best combined estimate, it was shown how a pair of combination coefficients can be calculated from estimates of the signal and noise variances of the respective region and boundary orientation estimates. These coefficients control the degree of the influence that either process has at a given iteration as the segmentation proceeds.

In the region-to-boundary interaction, the region based orientation estimate was in-

cluded as an extra term in the local cost for the boundary refinement. The relative weighting of the two energy terms was also determined by a pair of combination coefficients. Corner block detection and node insertion steps were described which are two instances where one process relies solely on the other to improve its segmentation. In the case of corner blocks the boundary process is able to overcome the inadequacies of the simple edge model of the region based orientation estimation. The node insertion step aims to improve the initial estimates from the boundary node selection and linking step, filling in gaps in the boundary graph using evidence gathered from the region process.

The chapter was concluded with a set of results for the cooperative segmentation on a range of synthetic and natural images. Two measures of performance were defined: pixel classification error and RMS boundary error, and were used to compare the results of the new method with the Spann and Wilson algorithm [88] [86]. The results showed the method to be good at segmentation of images which conformed to the image model, such as the foreground/background tasks, even in the presence of considerable amounts of noise. It was also shown to be better than previous methods at capturing corner features, which tend to be missed in the region-only methods such as [88].

There are a number of features of the segmentation method presented here which differentiate it from work reported elsewhere. The inhomogeneous block tessellation used by the region estimation process, which is used to produce the region adjacency graph, considerably reduces the computation of the iterative estimation and enables both local and global processing to be performed in a coherent manner. All parameter estimation is performed within the pyramid structure making the resulting method unsupervised. The orientation estimation process, although a sub-optimal adaptation of the quadtree estimation method originally proposed by Clippingdale [25], is nevertheless both computationally cheap and robust. The efficiency of this estimator is partly a consequence of the use of a set of new

small-size filtering kernels designed specifically for pyramid construction and orientation estimation [102]. The introduction of lateral processing further improves the estimates and helps the subsequent boundary processing. Finally, the orientation estimate is used as the vehicle for the novel cooperation strategy, which by exploiting the duality of the region and boundary representational structures, enables efficient region-boundary process interaction.

6.2 Limitations and Further Work

Although the results presented in Chapter 5 show the effectiveness of the method, there are a number of issues that need to be considered further. First there are some algorithmic considerations:

1. Orientation Estimation and Boundary Segmentation: In the present scheme, the enhancement of the orientation data and subsequent boundary segmentation are two separate processes. There is a feeling that these two stages, which both involve vertical and lateral processing, could be unified into one scheme to directly parallel the region processing. One of the benefits would be to reduce the overall computation.

2. Node Elimination and Addition: There is a need to make these steps more rigorous. Currently they have been formulated in heuristic terms. They are clearly two sides of the same coin and the decision criteria need to be made consistent. The experimental results confirm the belief, however, that the greater the noise the 'simpler' the boundary which can be supported by the data.

In addition there are, perhaps, more fundamental areas where extension and consolidation is required. These relate to the modification of the underlying image model (Chapter 2), and to the multiresolution feature description used:

3. Model Extensions: The segmentation scheme is limited by the simplicity of the underlying image model. As the results of Chapter 5 demonstrate, the homogeneous region model fails to deal adequately with the content of natural images. The boundary segmentation, however, is less affected by this because image boundaries are, by definition, invariant to variation across region interiors. To cope with this problem the region model has to be extended to allow for smooth variation in grey level across a neighbourhood. This requires modification of the scale selection and iterative estimation scheme, where simple averaging would no longer be possible. The model thus has to incorporate some measure of signal variance, as well as noise variance.

   In the modelling of boundaries, one obvious extension is to be able to represent line features as well as edges. The orientation pyramid which is used by the boundary process is constructed using operators which respond to edges only. Being able to differentiate between edges and lines would also help to make better decisions at the boundary linking stage.

   A conclusion from the results on synthetic images was the poor performance of the method on curved edges. This is a consequence of the use of a polyline boundary representation. An idea which was not investigated was to allow the boundary segments between nodes to be curved, i.e. quadratic arc segments or splines, which would result in a better fit to image curves. Such extensions of the boundary feature model from straight lines to higher order curves has been reported in [31].

4. Feature Description: A common property of regions found in images is texture. Texture feature extraction can be used to characterise the local region homogeneity, with a structural or statistical model. The hope would be that the current region model could be maintained by replacing the scalar, grey level data by a vector quantity which characterises the local texture [107] [109]. A generalisation of the model would allow both

variation of the region property, as described above, and texture to be simultaneously modelled.

Although the curves of the existing boundary model are defined by vertex points, the estimation algorithm does not detect vertex or branch features directly. In fact, the response of the orientation detection is null at a vertex. Again if the underlying feature description made such information available then this would further enhance the potential of the boundary segmentation. However, it must be noted that the sensitivity to noise of corner and branch features may limit the utility of this approach.

6.3 Concluding Remarks

This work has sought to unify a number of different ideas which have been used in image segmentation into a single framework. In particular, the problem of coherently combining both region and boundary information has been addressed. Multiresolution analysis has played a key role in the work, having an impact on the image model, representational structures and the computational efficiency of the resulting algorithms. What is apparent is that there are significant advantages to be had from such an approach.

One of the questions posed in the thesis introduction was to what degree *prior* information is necessary in low level vision tasks. The results presented here have shown that it is possible to begin to approach the capabilities of the HVS for certain segmentation tasks, by using only minimal assumptions about what is being seen. Indeed, one of the aims of this work has been to keep the image model simple in order to explore the question of the extent and form of the prior information required to achieve the desired result. However, as the above discussion noted, model simplicity does restrict the class of images which can be segmented.

There is clearly much work still to be done in the area of segmentation. Perhaps one of

the most fruitful ways forward for this work is to use some of the representational structures and algorithms with more generalised image descriptions such as the MFT. This would provide a rich image description from which grey level, texture, edge, line and branch feature information could be extracted and used to perform a segmentation. It is hoped that the work addressed in this thesis will provide a stepping stone for such ideas to be developed.

Appendix A

Derivation of M.M.S.E Filter

A.1 The MMSE Filter

The minimum mean square error filter is a fundamental result in filtering theory [74] [75] (also [58] [84] etc.). Given the arbitrary zero mean, random signals s and $\tilde{s}$ it is desired to obtain an estimate, $\hat{s}$ from the observation $\tilde{s}$ such that the mean square error

$$E[(s-\hat{s})^2] \tag{A.1}$$

is minimised. The best estimate of $\hat{s}$ is known [74] to be the conditional mean of s given $\tilde{s}$, that is

$$\hat{s} = E[s \mid \tilde{s}] \tag{A.2}$$

Equation (A.2) is in general difficult to solve as it is non-linear and the conditional probability density function $p_{s|\tilde{s}}$ is hard to calculate. The best *linear* estimate, however, is of the form

$$\hat{s}(x) = \int_{-\infty}^{\infty} h(u)\tilde{s}(x-u)du \tag{A.3}$$

where the filter impulse response $h(u)$ is determined such that the mean square error of equation (A.1) is minimised. Minimisation requires the orthogonality condition [75]

$$E[(s-\hat{s})\tilde{s}] = 0 \tag{A.4}$$

to be satisfied. Note that the linear estimate is also optimal for Gaussian processes.

A.2 A Constrained Approach

A tractable analytical solution for the shape of the filter can be found under the constraint that the filter is 1-d Gaussian of the form

$$h(x) = \alpha \exp(-\beta x^2) \tag{A.5}$$

and that the signal correlation function is also Gaussian

$$R_s(x) = \exp(-\rho x^2) \tag{A.6}$$

and the noise correlation is

$$R_n(x) = \sigma\delta(x) \tag{A.7}$$

A.2.1 Determining the Filter Parameters

Minimising (A.1) first with respect to α gives

$$2E[s(x) - \hat{s}(x)]\frac{\partial \hat{s}(x)}{\partial \alpha} = 0 \tag{A.8}$$

$$E[s(x)\frac{\partial \hat{s}(x)}{\partial \alpha}] = E[\hat{s}(x)\frac{\partial \hat{s}(x)}{\partial \alpha}] \tag{A.9}$$

Then substituting in $\hat{s}(x)$ from equation (A.3)

$$\int_{-\infty}^{\infty} \frac{\partial h(x)}{\partial \alpha} E[s(x)\tilde{s}(u-x)]dx = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(y)\frac{\partial h(x)}{\partial \alpha} E[\tilde{s}(u-x)\tilde{s}(u-y)]dxdy \tag{A.10}$$

$$\int_{-\infty}^{\infty} \frac{\partial h(x)}{\partial \alpha} R_s(x)dx = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(y)\frac{\partial h(x)}{\partial \alpha}\left[R_s(y-x) + \sigma\delta(y-x)\right]dxdy \tag{A.11}$$

Now $\frac{\partial h(x)}{\partial \alpha} = \exp(-\beta x^2)$ and $R_s(x) = \exp(-\rho x^2)$,

$$\int_{-\infty}^{\infty} \exp(-(\beta+\rho)x^2)dx =$$

$$\begin{aligned}
&\int_{-\infty}^{\infty} \alpha \exp(-\beta y^2) \int_{-\infty}^{\infty} \exp(-\beta x^2) \exp(-\rho(y-x)^2) dxdy \\
&+ \sigma \int_{-\infty}^{\infty} \alpha \exp(-\beta x^2) \exp(-\beta x^2) dx \\
= \quad & \alpha \left[\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \exp(-(\beta+\rho)(y^2+x^2) + 2\rho yx) dxdy \right. \\
& \left. + \sigma \int_{-\infty}^{\infty} \exp(-2\beta x^2) dx \right] \qquad \text{(A.12)}
\end{aligned}$$

Using the standard result for the definite exponential integral [1]

$$\int_0^{\infty} \exp(-ax^2) dx = \frac{1}{2}\sqrt{\frac{\pi}{a}},\ a > 0 \qquad \text{(A.13)}$$

on the first and last integrals and completing the square in the second integral, equation (A.12) becomes

$$\begin{aligned}
\sqrt{\frac{\pi}{\beta+\rho}} \quad = \quad & \alpha \left[\int_{-\infty}^{\infty} \exp(-(\beta+\rho)x^2 + (\frac{\rho^2 x^2}{\beta+\rho})) \cdot \right. \\
& \left. \int_{-\infty}^{\infty} \exp(-(\beta+\rho)y^2 + 2\rho xy - (\frac{\rho^2 x^2}{\beta+\rho})) dxdy + \sigma\sqrt{\frac{\pi}{2\beta}} \right] \qquad \text{(A.14)}
\end{aligned}$$

$$\begin{aligned}
= \quad & \alpha \left[\int_{-\infty}^{\infty} \exp(-\frac{\beta(\beta+2\rho)}{\beta+\rho} x^2) \cdot \right. \\
& \left. \int_{-\infty}^{\infty} \exp(-(\sqrt{\beta+\rho}y - \frac{\rho x}{\sqrt{\beta+\rho}})^2) dxdy + \sigma\sqrt{\frac{\pi}{2\beta}} \right] \qquad \text{(A.15)}
\end{aligned}$$

The second integral is evaluated by the substitution $u = \sqrt{\beta+\rho}y - \frac{\rho x}{\sqrt{\beta+\rho}}, du = dy\sqrt{\beta+\rho}$, and the standard result of equation (A.13) yielding

$$\sqrt{\frac{\pi}{\beta+\rho}} = \alpha \left[\sqrt{\frac{\pi}{\beta+\rho}} \int_{-\infty}^{\infty} \exp(-\frac{\beta(\beta+2\rho)}{\beta+\rho} x^2) dx + \sigma\sqrt{\frac{\pi}{2\beta}} \right] \qquad \text{(A.16)}$$

And re-applying equation (A.13) to the remaining integral gives

$$\sqrt{\frac{\pi}{\beta+\rho}} = \alpha \left[\sqrt{\frac{\pi}{\beta+\rho}} \sqrt{\frac{\pi(\beta+\rho)}{\beta(\beta+2\rho)}} + \sigma\sqrt{\frac{\pi}{2\beta}} \right] \qquad \text{(A.17)}$$

After some manipulation it can be shown that

$$\alpha = \sqrt{\frac{2\beta(\beta+2\rho)}{(\beta+\rho)}} \frac{1}{(\sqrt{2\pi} + \sigma\sqrt{\beta+2\rho})} \qquad \text{(A.18)}$$

It is clear that the parameter α is dependent on β which is the value determining the variance of the filter. Persevering with the analysis, however, to determine an expression for β does produce a useful result.

Differentiating equation (A.1) with respect to β this time, following the procedures of equations (A.8) to (A.11)

$$2E[s(x) - \hat{s}(x)]\frac{\partial \hat{s(x)}}{\partial \beta} = 0 \tag{A.19}$$

$$E[s(x)\frac{\partial \hat{s}(x)}{\partial \beta}] = E[\hat{s}(x)\frac{\partial \hat{s}(x)}{\partial \beta}] \tag{A.20}$$

$$\int_{-\infty}^{\infty} \frac{\partial h(x)}{\partial \beta} E[s(x)\tilde{s}(u-x)]dx = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(y)\frac{\partial h(x)}{\partial \beta} E[\tilde{s}(u-x)\tilde{s}(u-y)]dxdy \tag{A.21}$$

$$\int_{-\infty}^{\infty} \frac{\partial h(x)}{\partial \beta} R_s(x)dx = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(y)\frac{\partial h(x)}{\partial \beta}\left[R_s(y-x) + \sigma\delta(y-x)\right]dxdy \tag{A.22}$$

Now $\frac{\partial h(x)}{\partial \beta} = -\alpha \exp(-\beta x^2)$ and $R_s(x) = \exp(-\rho x^2)$,

$$\begin{aligned}\int_{-\infty}^{\infty} x^2 \exp(-(\beta+\rho)x^2)dx = & \\ = \quad \alpha \Bigg[\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^2 \exp(-(\beta+\rho)(y^2+x^2) + 2\rho yx)dxdy & \\ + \sigma \int_{-\infty}^{\infty} x^2 \exp(-2\beta x^2)dx\Bigg] & \end{aligned} \tag{A.23}$$

Using the standard result for the definite exponential integral [1]

$$\int_0^{\infty} x^2 \exp(-ax^2)dx = \frac{1}{4}\sqrt{\frac{\pi}{a}} \cdot \frac{1}{a},\ a > 0 \tag{A.24}$$

on the first and last integrals and completing the square in the second integral, equation (A.23) becomes

$$\begin{aligned}\sqrt{\frac{\pi}{\beta+\rho}}\ \frac{1}{2(\beta+\rho)} = & \\ \alpha \Bigg[\int_{-\infty}^{\infty} x^2 \exp(-\frac{\beta(\beta+2\rho)}{\beta+\rho}x^2) \cdot & \\ \int_{-\infty}^{\infty} \exp(-(\sqrt{\beta+\rho y} - \frac{\rho x}{\sqrt{\beta+\rho}})^2)dxdy + \sigma\sqrt{\frac{\pi}{2\beta}}\ \frac{1}{4\beta}\Bigg] & \end{aligned} \tag{A.25}$$

The second integral is evaluated by substitution (c.f equation (A.16)), and using the standard result of equation (A.13), giving

$$\sqrt{\frac{\pi}{\beta+\rho}}\,\frac{1}{2(\beta+\rho)} = \alpha\left[\sqrt{\frac{\pi}{\beta+\rho}}\cdot\sqrt{\frac{\pi(\beta+\rho)}{\beta(\beta+2\rho)}}\cdot\frac{\beta+\rho}{2\beta(\beta+2\rho)}+\sigma\sqrt{\frac{\pi}{2\beta}}\,\frac{1}{4\beta}\right] \tag{A.26}$$

After some manipulation it can be shown that

$$\begin{aligned}(\beta-\rho)(\beta+2\rho)^{\frac{3}{2}} &= \frac{2\rho^2}{\sigma}\sqrt{2\pi}\\ \Rightarrow \beta &= \rho+\frac{2\rho^2\sqrt{2\pi}}{\sigma(\beta+2\rho)^{\frac{3}{2}}}\end{aligned} \tag{A.27}$$

Given ρ and σ, β can be determined numerically [50] using equation (A.27) and hence α from equation (A.18).

Appendix B

Optimal Lowpass and Edge Kernel Designs

This paper has been accepted for publication for IEEE, Transactions on PAMI [99].

Appendix C

Conference Paper

This paper was presented at the 7th. SCIA, Aalborg, Denmark, 1991 [11].

References

[1] M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions.* Dover Publications Inc., New York, 1972.

[2] A. A. Amini, T. E. Weymouth, and R. Jain. Using Dynamic Programming for Solving Variational Problems In Vision. *IEEE Trans. PAMI.*, 12(9):855–867, 1990.

[3] R. Bajcsy. Active Perception. *Proc. IEEE, Special Issue on Computer Vision*, pages 966–1005, 1988.

[4] D. Ballard. Generalising the Hough Transform. *Pattern Recognition*, 13:111–22, 1981.

[5] D. H. Ballard and C. M. Brown. *Computer Vision.* Prentice Hall, New Jersey, 1982.

[6] M. F. Barnsley. *Fractals Everywhere.* Academic Press, 1988.

[7] F. Bergholm. Edge Focusing. *IEEE Trans. PAMI.*, 9(6):726–741, 1987.

[8] J. Besag. Spatial Interaction and the Statistical Analysis of Lattice Systems (with discussion). *Journal of the Royal Statistical Society*, 36(Series B):192–236, 1974.

[9] J. Besag. On the Statistical Analysis of Dirty Pictures. *Journal of the Royal Statistical Society*, 48(3):259–302, 1986.

[10] J. Besag. Towards Bayesian Image Analysis. In J. G. McWhirter, editor, *Mathematics In Signal Processing II.* Clarendon Press, Oxford, 1990.

[11] A. Bhalerao and R. Wilson. Multiresolution Image Segmentation Combining Region and Boundary Information. In *Proc. 7th Scandinavian Conf. Im. Analysis*, pages 1162–1169, Aalborg, Denmark, 1991.

[12] J. Bigün. *Local Symmetry Features in Image Processing.* PhD thesis, University of Linköping, Sweden, 1988.

[13] J. Bigün. Frequency and Orientation Selective Texture Measures Using Linear Symmetry and Laplacian Pyramid. In *Proc. SPIE Conf. Visual Comms. and Image Processing*, pages 1319–1331, Lausanne, Switzerland, 1990.

[14] C. Blakemore and F. W. Campbell. On the Existence in the Human Visual System of Neurons Selectively Sensitive to the Orientation and Size of Retinal Images. *Journal of Physiology*, 203:237–60, 1969.

[15] P. Bonnin and B. Zavidovique. An Edge PointRegion Cooperative Segmentation Specific to 3D Scene Reconstruction Application. In *Proc. SPIE Conf. Visual Comms. and Image Processing*, pages 1270–1281, Lausanne, Switzerland, 1990.

[16] C. Bouman and B. Liu. Multiple Resolution Segmentation of Textured Images. *IEEE Trans. PAMI.*, 13:99–113, 1991.

[17] J. M. Du Buf, M. Kardan, and M. Spann. Texture Feature Performance for Image Segmentation. *Pattern Recognition*, 23(3/4):291–309, 1990.

[18] P. J. Burt and E. H. Adelson. The Laplacian Pyramid as a Compact Image Code. *IEEE Trans. Comp.*, COM-31:532–540, 1983.

[19] P. J. Burt, T. H. Hong, and A. Rosenfeld. Segmentation and Estimation of Image Region Properties Through Cooperative Hierarchical Computation. *IEEE Trans. Sys, Man Cyber.*, 11(12):802–809, 1981.

[20] A. Calway. *The Multiresolution Fourier Transform: A general Purpose Tool for Image Analysis.* PhD thesis, Department of Computer Science, The University of Warwick, UK, September 1989.

[21] A. Calway and R. G. Wilson. A Unified Approach to Feature Extraction Based on an Invertible Image Transform. In *Proc. 3rd IEE Int. Conf. Image Processing*, pages 651–655, Warwick, U.K, 1989.

[22] J. Canny. A Computational Approach to Edge Detection. *IEEE Trans. PAMI.*, 8:679–698, 1986.

[23] R. Chellappa and R. L. Kashyap. Digital Image Restoration using Spatial Interaction Models. *IEEE Trans. Acous. Speech Sig. Proc.*, 30:461–472, 1982.

[24] P. C. Chen and T. Pavlidis. Image Segmentation as an Estimation Problem. In A. Rosenfeld, editor, *Image Modelling*, pages 9–28. Academic Press, 1981.

[25] S. Clippingdale. *Multiresolution Image Modelling and Estimation.* PhD thesis, Department of Computer Science, The University of Warwick, UK, September 1988.

[26] S. C. Clippingdale and R. Wilson. Quad-Tree Image Estimation: A New Image Model and its Application to MMSE Image Restoration. In *Proc. 5th Scandinavian Conf. Image Analysis*, pages 699–706, Stockholm, Sweden, 1987.

[27] S. C. Clippingdale and R. Wilson. Least Squares Estimation on a Multiresolution Pyramid. In *Proc. ICASSP-89*, pages 1409–12, Glasgow, 1989.

[28] P. Cohen and H. H. Nguyen. Unsupervised Bayesian Estimation for Segmenting Textured Images. In *Proc. Int. Conf. Computer Vision*, pages 303–309, 1988.

[29] I. Daubechies. The Wavelet Transform, Time-Frequency Localisation and Signal Analysis. *IEEE Trans. Information Theory*, 36:961–1005, 1990.

[30] A. Davies and R. Wilson. Linear Feature Extraction Using the Multiresolution Fourier Transform. Technical Report RR170, Department of Computer Science, University of Warwick, 1990.

[31] A. Davies and R. Wilson. Curve and Corner Feature Extraction using the Multiresolution Fourier Transform. Technical report, Department of Computer Science, University Of Warwick, Coventry, U.K., 1991.

[32] R. L. De Valois, D. G. Albrecht, and L. G. Thorell. Spatial Frequency Selectivity of Cells in Macaque Visual Cortex. *Vision Research*, 22:545–59, 1982.

[33] P. A. Devijver and J. Kittler. *Pattern Recognition a Statistical Approach*. Prentice-Hall International, Inc., London, 1982.

[34] R. O. Duda and P. E. Hart. Use of the Hough Transform to Detect Lines and Curves in Pictures. *Communications of the ACM*, 15:11–15, 1972.

[35] K. S. Fu, R. C. Gonzalez, and C. S. G. Lee. *Robotics: Control, Sensing, Vision, and Intelligence*. Industrial Engineering Series. McGraw-Hill Inc., 1987.

[36] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, 1972.

[37] P. W. Fung, G. Grebbin, and Y. Attikiouzel. Contextual Classification and Segmentation of Textured Images. In *Proc. ICASSP-90*, pages 2329–2332, Albuquerque, 1990.

[38] D. Geman, S. Geman, C. Graffigne, and P. Dong. Boundary Detection by Constrained Optimisation. *IEEE Trans. PAMI.*, 12:609–28, 1990.

[39] S. Geman and D. Geman. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Trans. PAMI.*, 6(6):721–741, 1984.

[40] B. Gidas. A Renormalization Group Approach to Image Processing Problems. *IEEE Trans. PAMI.*, 11(2):164–180, 1989.

[41] R. C. Gonzalez and P. Wintz. *Digital Image Processing (2nd. Ed)*. Addison-Wesley, New York, 1987.

[42] G. H. Granlund. In Search of a General Picture Processing Operator. *Computer Vision, Graphics and Image Processing*, 8:155–173, 1978.

[43] J. F. Haddon and J. F. Boyce. Simultaneous Image Segmentation and Edge Detection. In *Proc. IEE 3rd Int. Conf. on Image Proc. and its Applications*, pages 411–415, University of Warwick, U. K., 1989.

[44] E. R. Hancock and J. Kittler. Discrete Relaxation. *Pattern Recognition*, 23:711–33, 1990.

[45] A. R. Hanson and Riseman. Segmentation of Natural Scenes. In *Computer Vision Systems*. Academic Press, 1978.

[46] R. M. Harlick. Zero-Crossing of Second Directional Derivative Edge Operator. *IEEE Trans. PAMI.*, 6:58–68, 1978.

[47] R. M. Harlick. Image Segmentation A Survey. In O. D. Faugeras, editor, *Fundamentals in Computer Vision*, pages 210–223. Cambridge University Press, 1983.

[48] T. H. Hong and A. Rosenfeld. Compact Region Extraction Using Weighted Pixel Linking in a Pyramid. *IEEE Trans. PAMI.*, 2(12):222–29, 1984.

[49] T. H. Hong and M. Shneier. Extracting Compact Objects Using Linked Pyramids. *IEEE Trans. PAMI.*, 6:229–237, 1984.

[50] R. J. Hosking, D. C. Joyce, and J. C. Turner. *First Steps In Numerical Analysis*. Hodder and Stoughton Ltd., 1978.

[51] D. H. Hubel. *Eye, Brain and Vision*. New York, Scientific American Library, 1988.

[52] D. H. Hubel and T. N. Wiesel. Brain Mechanisms of Vision. *Scientific American*, pages 130–144, 1979.

[53] M. F. Hueckel. An Operator Which Locates Edges in Digitised Pictures. *Journal of the ACM*, 18:113–125, 1971.

[54] R. A. Hummel and S. W. Zucker. On the Foundations of Relaxation Labeling Processes. *IEEE Trans. PAMI.*, 5(36):267–287, 1983.

[55] G. M. Hunter and K. Steiglitz. Operations on Images Using Quad Trees. *IEEE Trans. PAMI.*, 1:145–53, 1979.

[56] J. Illingworth and J. Kittler. A Survey of the Hough Transform. *Computer Vision, Graphics and Image Processing*, 44:87–116, 1988.

[57] A. E. Jacquin. A Novel Fractal Block-Coding Technique for Digital Images. In *Proc. ICASSP-90*, pages 2225–2228, Albuquerque, 1990.

[58] A. K. Jain. *Fundamentals of Digital Image Processing.* New York, Addison-Wesley, 1989.

[59] J. Kittler and J. Föglein. A General Contextual Classification Method for Segmentation. In *Proc. 3th Scandinavian Conf. Im. Analysis*, pages 90–95, Copenhagen, Denmark, 1983.

[60] D. E. Knuth. *Seminumerical Algorithms (2nd. Ed.).* Academic Press, 1981.

[61] H. Knutsson. *Filtering and Reconstruction in Image Processing.* PhD thesis, University of Linköping, Sweden, 1982.

[62] H. E. Knutsson, R. Wilson, and G. H. Granlund. Estimation the Local Orientation of Anisotropic 2-d Signals. In *Proc. IEEE ASSP Workshop on Spect. Est.*, pages 234–9, Tampa, 1983.

[63] J. Konrad and E. Dubois. Multigrid Bayesian Estimation of Image Motion Fields Using Stochastic Relaxation. In *Proc. Int. Conf. Computer Vision*, pages 354–361, 1988.

[64] S. Levialdi. Basic Ideas for Image Segmentation. In O. D. Faugeras, editor, *Fundamentals in Computer Vision*, pages 239–261. Cambridge University Press, 1983.

[65] S. Levialdi. Edge Extraction Techniques. In O. D. Faugeras, editor, *Fundamentals in Computer Vision*, pages 118–144. Cambridge University Press, 1983.

[66] Y. T. Liow and T. Pavlidis. Use of Shadows for Extracting Buildings in Aerial Images. *Computer Vision, Graphics and Image Processing*, pages 242–277, 1990.

[67] S. G. Mallat. A Theory for Multiresolution Signal Decomposition: the Wavelet Representation. *IEEE Trans. PAMI.*, 11:674–93, 1989.

[68] D. Marr. *Vision.* Freeman Press, San Fransisco, C.A., 1982.

[69] D. Marr and E. Hildreth. Theory of Edge Detection. *Proceedings of the Royal Society London*, B207:187–217, 1980.

[70] P. Meer, E. S. Baugher, and A. Rosenfeld. Frequency Domain Analysis and Synthesis of Pyramid Generating Kernels. *IEEE Trans. PAMI.*, 9:512–22, 1987.

[71] J. L. Melsa and D. L. Cohn. *Decision and estimation theory.* McGraw-Hill, New York, 1978.

[72] A. Montanvert, P. Meer, and A. Rosenfeld. Hierarchial Image Analysis Using Irregular Tessellations. *IEEE Trans. PAMI.*, 13:307–315, 1991.

[73] D. W. Murray and B. F. Buxton. Scene Segmentation from Visual Motion Using Global Optimisation. *IEEE Trans. PAMI.*, 9:220–228, 1987.

[74] A. Papoulis. *Signal Analysis.* McGraw-Hill, New York, 1977.

[75] A. Papoulis. *Probability, Random Variables and Stochastic Processes (2nd Ed.).* McGraw-Hill, New York, 1984.

[76] P. Parent and S. W. Zucker. Trace Inference, Curvature Consistency, and Curve Detection. *IEEE Trans. PAMI.*, 11(8):823–839, 1989.

[77] T. Pavlidis and Y. T. Liow. Integrating Region Growing and Edge Detection. Technical Report TR.89.01.05, Department of Computer Science, State University of New York, Stony Brook, 1989.

[78] J. Phillips. *The NAG Library: A Beginners Guide.* Clarendon Press, Oxford, 1986.

[79] W. K. Pratt. *Digital Image Processing.* Wiley, New York, 1983.

[80] J. Princen, J. Illingworth, and J. Kittler. A Hierarchical Approach to Line Extraction Based on the Hough Transform. *Computer Vision, Graphics and Image Processing*, 52:57–77, 1990.

[81] L. Roberts. Machine Perception of 3-Dimensional Solids. In J. Tippett, editor, *Optical and Electrooptical Information Processing.* MIT Press, Cambridge MA, 1965.

[82] A. Rosenfeld. Image Analysis: Problems, Progress and Prospects. *Pattern Recognition*, 17:3–12, 1984.

[83] A. Rosenfeld and M. Thurston. Edge and Curve Detection for Visual Scene Analysis. *IEEE Trans. Comp.*, 5(20):562–569, 1971.

[84] R. J. Schalkoff. *Digital Image Processing and Computer Vision.* New York, Wiley, 1989.

[85] T. Simchony, R. Chellappa, and Z. Lichtenstein. Graduated Nonconvexity Algorithm for Image Estimation Using Computed Gauss Markov Field Models. In *Proc. ICASSP-89*, pages 1417–1420, Glasgow, 1989.

[86] M. Spann. *Texture Description and Segmentation in Image Processing.* PhD thesis, Department of Electrical Engineering, The University of Aston in Birmingham, UK, September 1985.

[87] M. Spann and C. Horne. Image Segmentation Using a Dynamic Thresholding Pyramid. *Pattern Recognition*, 22(6):729–732, 1989.

[88] M. Spann and R. G. Wilson. A Quad-Tree Approach to Image Segmentation Which Combines Statistical and Spatial Information. *Pattern Recognition*, 18(3/4):257–269, 1985.

[89] B. Stourstrup. *The C++ Programming Language.* Addeison Wesley, Reading, MA, 1986.

[90] S. L. Tanimoto and T. Pavlidis. A Hierarchical Data Structure for Picture Processing. *Computer Vision, Graphics and Image Processing*, pages 104–119, 1975.

[91] D. Terzopoulos. Image Analysis Using Multrigird Relaxation Methods. *IEEE Trans. PAMI.*, 8:129–139, 1986.

[92] C. W. Therrien. *Decision, Estimation and Classification.* J. Wiley and Sons, 1989.

[93] M. Todd. *Image Data Compression Based on a Multiresolution Signal Model.* PhD thesis, Department of Computer Science, The University of Warwick, UK, November 1989.

[94] M. Todd and R. Wilson. An Anisotropic Multiresolution Image Data Compression Algorithm. In *Proc. ICASSP-89*, pages 1969–72, Glasgow, 1989.

[95] P. J. M. van Laarhoven and E. H. L. Aarts. *Simulated Annealing: Theory and Applications.* D. Reidel Publishing, Dordrecht, Holland, 1987.

[96] F. M. Wahl. *Digital Image Signal Processing.* Artech House, 1987.

[97] J. Waite and M. Beaumont. An Introduction to Block Based Fractal Image Coding. Technical report, British Telecom Research Laboratories, 1991.

[98] R. Watt. *Visual Processing: Computational, Psychophysical and Cognitive Research.* Lawrence Erlbaum Associates, Publishers, 1988.

[99] R. Wilson and A. H. Bhalerao. Kernel Designs for Efficient Multiresolution Edge Detection and Orientation Estimation. *Accepted for publication, IEEE Trans. PAMI*, 1991.

[100] R. Wilson, A. D. Calway, and E. R. S. Pearson. A Generalized Wavelet Transform for Fourier Analysis: the Multiresolution Fourier Transform and its Application to Image and Audio Signal Analysis. *IEEE Trans. IT, Special Issue on Wavelet Representations*, 1992.

[101] R. Wilson and S. C. Clippingdale. A Class of Nonstationary Image Models and their Applications. In J. G. McWhirter, editor, *Mathematics In Signal Processing II.* Clarendon Press, Oxford, 1990.

[102] R. Wilson, S. C. Clippingdale, and A. H. Bhalerao. Robust Estimation of Local Orientations in Images Using a Multiresolution Approach. In *Proc. 5rd SPIE Conf. Vis. Comm. Image Proc.*, Lausanne, 1990.

[103] R. Wilson, S. C. Clippingdale, and M. P. Todd. Multiresolution Image Modelling and its Applications. *To be published*, 1990.

[104] R. Wilson and G. H. Granlund. The Uncertainty Principle in Image Processing. *IEEE Trans. PAMI.*, 6:758–67, 1984.

[105] R. Wilson and H. Knutsson. Uncertainty and Visual Inference. *IEEE Trans. Sys, Man Cyber.*, 18:302–312, 1988.

[106] R. Wilson, H. E. Knutsson, and G. H. Granlund. Anisotropic Nonstationary Image Estimation and its Applications: Part II - Predictive Image Coding. *IEEE Trans. Comp.*, 31:398–406, 1983.

[107] R. Wilson and M. Spann. Finite Prolate Spheroidial Sequence and Their Applications I, II: Image Feature Description and Segmentation. *IEEE Trans. PAMI.*, 10:193–203, 1988.

[108] R. Wilson and M. Spann. A New Approach To Clustering. *Pattern Recognition*, 23(12):1413–1425, 1990.

[109] R. G. Wilson and M. Spann. *Image Segmentation and Uncertainty.* Pattern Recognition and Image Processing Series. Research Studies Press Ltd, 1988.

[110] A. P. Witkin. Scale-Space Filtering. In *Proc. IJCAI*, Karlsruhe, 1983.

[111] J. W. Woods. Two-Dimensional Discrete Markovian Fields. *IEEE Trans. Information Theory*, 18:232–240, 1972.

[112] S. W. Zucker. A Biologically Motivated Approach to Early Visual Computations: Orientation, Texture and Optical Flow. In J. Kittler, editor, *Proc. 4th BPRA Conf. on PR, Lecture Notes in CS*, volume 301, pages 417–428. Springer-Verlag, 1988.